

An Automated Anti-Cheat Framework for Online Exams in Pakistan



Author

Taskeen Fatima

FALL 2018-MS-18(CSE) 00000274101

MS-18 (CSE)

Supervisor

Dr. Farooque Azam

DEPARTMENT OF COMPUTER & SOFTWARE ENGINEERING COLLEGE
OF ELECTRICAL & MECHANICAL ENGINEERING
NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY
ISLAMABAD

AUG 2022

An Automated Anti-Cheat Framework for Online Exams in Pakistan

Author

Taskeen Fatima

FALL 2018-MS-18(CSE) 00000274101

A thesis submitted in partial fulfillment of the requirements for the degree of
MS Software Engineering

Thesis Supervisor:

Dr. Farooque Azam

Thesis Supervisor's Signature: _____

DEPARTMENT OF COMPUTER & SOFTWARE ENGINEERING
COLLEGE OF ELECTRICAL & MECHANICAL ENGINEERING
NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY,
ISLAMABAD

AUG 2022

DECLARATION

I certify that this research work titled “*An Automated Anti-Cheat Framework for Online Exams in Pakistan*” is my own work under the supervision of Dr. Farooque Azam. This work has not been presented elsewhere for assessment. The material that has been used from other sources has been properly acknowledged / referred.

Signature of Student

Taskeen Fatima

FALL 2018-MS-18(CSE) 00000274101

LANGUAGE CORRECTNESS CERTIFICATE

This thesis is free of typing, syntax, semantic, grammatical and spelling mistakes. Thesis is also according to the format given by the University for MS thesis work.

Signature of Student

Taskeen Fatima

FALL 2018-MS-18(CSE) 00000274101

Signature of Supervisor

COPYRIGHT STATEMENT

- Copyright in text of this thesis rests with the student author. Copies (by any process) either in full, or of extracts, may be made only in accordance with instructions given by the author and lodged in the Library of NUST College of E&ME. Details may be obtained by the Librarian. This page must form part of any such copies made. Further copies (by any process) may not be made without the permission (in writing) of the author.
- The ownership of any intellectual property rights which may be described in this thesis is vested in NUST College of E&ME, subject to any prior agreement to the contrary, and may not be made available for use by third parties without the written permission of the College of E&ME, which will prescribe the terms and conditions of any such agreement.
- Further information on the conditions under which disclosures and exploitation may take place is available from the Library of NUST College of E&ME, Rawalpindi

ACKNOWLEDGEMENTS

I am extremely thankful to Allah Almighty for his bountiful blessings throughout this work. Indeed this would not have been possible without his substantial guidance through each and every step, and for putting me across people who could drive me through this work in a superlative manner. Indeed none be worthy of praise but the Almighty.

I am profusely grateful to my beloved parents for their love, prayers, support and sacrifices for educating and preparing me for my future. I also thank my siblings who encouraged me and prayed for me throughout the time of my research.

I would also like to express my gratitude to my supervisor **Dr. Farooque Azam** and my co-supervisor, **Dr. Wasi Haider**, for their constant motivation, patience, enthusiasm, and immense knowledge. Their guidance helped me throughout my research and writing of this thesis. I could not have imagined having a better advisor and mentor for my MS study.

I would like to pay special thanks to **Muhammad Waseem Anwar & Abdul Wahab Muzaffar** for his incredible cooperation and providing help at every phase of this thesis. He has guided me and encouraged me to carry on and has contributed to this thesis with a major impact. Also, this research study is conducted in collaboration of a project “An open-source online examination system” at Saudi Electronic University

I would also like to thank my Guidance Committee Members **Dr. Muhammad Umar** and **Dr. Arsalan Shaukat** for being on my thesis guidance and evaluation committee. Their recommendations are very valued for improvement of the work.

Last but not the least, I would like to express my gratitude to all the individuals who have rendered valuable assistance to my study.

Thanks for all your encouragement!

Dedicated to my beloved parents whose tremendous support and cooperation led me to this wonderful accomplishment.

ABSTRACT

In the past few decades E-learning in higher education is increased and play vital role in pandemic like COVID-19. Particularly, online examinations are conducted on e-learning platforms that leads to save time and other resources for both institutes and students. Also, where online examination has advantages but have impact of dis honesty of students in online examination. For this reason, numerous research is available that proposes methodologies and techniques for seamless execution of online examination and prevent the cheating in online examination. It is very important to have adequate solution for above problem. However, this study provides the comparative analysis of online examination techniques, tools performed on 50 studies from last five years 2017 to 2021. All the studies identified the lack of framework for automated online examination system for less developed countries. Furthermore, frameworks proposed, and the industrial or commercial tool do not provide the cost-effective, offline monitoring solution with less requirement of internet bandwidth which is very important for less developed countries to adopt the OEP systems. Hence there is a sheer need of developing an anti-cheat framework for less developed countries that predicts anti-cheating behavior of students as well as cost effective solution.

Keeping this in view, an open-source framework for anti-cheat online examination in Pakistan has proposed that ensures improved ease of use of software and cheating detecting during offline monitoring and low network bandwidth. The framework also automated the cheating prediction reports. The proposed framework used NLP technique to find out the cheating prediction on student's monitoring process and data. The benefit of using the NLP technique to generate the high accurate results as per studies. Now to make the cost-effective solution the framework is developed using cloud technology that also help to monitor the offline processes or activities and ultimately provide the usability of minimum internet bandwidth requirement. The framework not only easy to use for students but also provide the improved usability for instructors are administrators who are authorized to generate the automated reports. Furthermore, we validate the framework by conducting the exam for two different subjects from 11 students. Our framework successfully analyzed the cheating and non-cheating cases by achieving the 90% accuracy.

Keywords: Online Exam Proctoring, Distance Education, Machine Learning, Online examination dis-honesty, Cheating Prediction in Online Examination, Cloud Computing

TABLE OF CONTENTS

DECLARATION	1
LANGUAGE CORRECTNESS CERTIFICATE	2
COPYRIGHT STATEMENT	3
ACKNOWLEDGEMENTS	4
ABSTRACT	6
TABLE OF CONTENTS	7
TABLE OF FIGURES	9
LIST OF TABLES	11
CHAPTER 1: INTRODUCTION	13
1.1. Background Study	13
1.2. Problem Statement	17
1.3. Proposed Methodology	17
1.4. Research Contribution.....	19
1.5. Thesis Organization	19
CHAPTER 2: LITERATURE REVIEW	22
2.1. Systematic Literature Review	22
2.2. Research Gap	46
CHAPTER 3: PROPOSED METHODOLOGY	49
3.1. Solution Idea	49
3.2. Proposed System Workflow	50
CHAPTER 4: IMPLEMENTATION	65
4.1. Automated Online Cheating Prevention System.....	65
4.2. Client-Side Implementation	66
4.2.7. Network Infrastructure Handling	71
4.3. Server-Side Implementation	71
4.3.1. Anaconda Python	71
4.3.2. Fetch Dataset files & Utilize	72
4.3.3. Algorithm.....	72
4.4. Tool Interface.....	72
CHAPTER 5: VALIDATION	78
5.1. Validation Process.....	78

5.2.	Test Project Details	78
CHAPTER 6: Discussion and Limitation.....		88
6.1.	Discussion	88
6.2.	Limitation.....	89
CHAPTER 7: CONCLUSION AND FUTURE WORK		91
REFERENCES		92

TABLE OF FIGURES

Figure 1- Problem Statement Summary	17
Figure 2- Research Flow	18
Figure 3 - Thesis Outline.....	20
Figure 4- Overview of review process	23
Figure 5- Search Process Flow.....	27
Figure 6- Proposed System Workflow	50
Figure 7-Cloud Based Architecture.....	51
Figure 8-Client Application Flow	51
Figure 9-User Authentication Flow.....	53
Figure 10-Question Bank Generation Flow	53
Figure 11- Key Logging Flow.....	54
Figure 12- Application History Monitoring.....	55
Figure 13- ETL Data Generation Approach.....	56
Figure 14- Data Cleaning and Pre-processing.....	58
Figure 15- Formula for Probabilities.....	62
Figure 16- Architecture Diagram for Tools and Techniques	65
Figure 17-.Net Framework and Design Some major features.....	66
Figure 18- Dataset Generated files.....	67
Figure 19- User Authentication Flow.....	68
Figure 20- Question Bank Generation Flow	69
Figure 21-Key Logs Generation Flow	69
Figure 22- Application History Monitoring.....	70
Figure 23- Fetch Dataset File & Utilization.....	72
Figure 24- Client-Side Application Interface.....	73
Figure 25- Application Interface II	73
Figure 26- Cloud Server Data Folders	74
Figure 27-Python Script Folder.....	74
Figure 28- Executed Python Script	75
Figure 29- Automated Cheating Prediction Report on Cloud Server	75
Figure 30- Network Connectivity Handling.....	84

Figure 31- Internet Connectivity Function..... 84
Figure 32- Precision, Recall & Accuracy Formula..... 85

LIST OF TABLES

Table 1- Summary of Search Terms and Corresponding Results.....	28
Table 2- Summary of research papers based on publication year.....	29
Table 3- Summary of research papers based on scientific database.....	30
Table 4- Summary of research papers based on scientific database and Journal & Conference Papers.....	30
Table 5- Data Extraction & Synthesis.....	31
Table 6- Synthesis Results - OEM Categories.....	33
Table 7- OEM Techniques & Approaches.....	36
Table 8- Machine Learning Techniques.....	38
Table 9- Automated OEM Tools.....	40
Table 10- Automated Proposed Solution Dataset.....	41
Table 11- Automated OEP Comparison of Approaches.....	45
Table 12- Cheating Prediction Weightage Distribution.....	63
Table 13- Exam Project Sample Details.....	78
Table 14- Non-Cheating Case Results.....	80
Table 15- Cheating Case Results.....	83
Table 16- Prediction Results for Cheating and Non-Creating Cases.....	85
Table 17- Total Prediction Results in Percentage.....	86

Chapter 1

Introduction

CHAPTER 1: INTRODUCTION

This section provides a detail introduction about the important concepts related to our research, the current problem and an overview of our solution. It is organized into five sub sections. **Section 1.1** describes the background study, **Section 1.2** provides the problem statement of research, **Section 1.3** discusses the proposed methodology, **Section 1.4** gives the detail about research contribution, and thesis organization is presented in **Section 1.5**

1.1. Background Study

The purpose of this section is to introduce the background study of multiple important concepts which has been used in this research. These concepts include:

- Online Exam Proctoring
 - What is online exam proctoring
 - Importance of Online exam proctoring in COVID19
 - Characteristics of Online exam proctoring
- Types of Online Exam Proctoring
 - Types of Online exam proctoring
 - Importance of Automated online exam proctoring
 - Important Techniques of Automated online exam proctoring

1.1.1. Online Exam Proctoring

Importance of OEP in COVID19

There is no doubt that online learning has gained lot of popularity in the pandemic (COVID19) where the term online exam is not surprising and most of the time known to everyone due to circumstances. The importance of the OEP in pandemic raises as those institutes which were following the traditional methods of face-to-face teaching and conducting exams faced a lot of issues and had to adopt the new era of OEP.

The importance of OEP can be recognized by its effective facilities to the institutes in pandemic. All those institutes who followed the traditional methods of teaching and exam conduction had to follow the new OEP model. However, it is a fact that online learning implementation has some challenges which vary from country to country. Here it is important to

highlight that all the challenges are particularly to those country which are under development. As we all know that E-learning has some serious nature deficiencies in conforming the effective mechanism, authentication of the user while conducting the exam a throughout the session but eventually the only solution that we have solutions in pandemic is online exam proctoring. Due to high cost of hardware or devices, software complexity in integration, High accuracy, and other factors this solution is hard to adopt for under developing countries.

1.1.2. Types of Online Exam Proctoring (OEP)

As explained in above section about the OEP and its importance, it's a solution that provides online exam conduction with all the necessary needs and allow institutes to survive in pandemic and continuing the process of exam conduction with more ease. Most of the developed countries has shifted their e-learning models and exam conduction through OEP and those countries which are less developed are adopting the OEP solution according to their needs and financial factors.

Types of Online Exam proctoring

There are essentially three main types of online exam proctoring.

- 1) Fully live Online Proctoring:** Live proctoring is a type of service provide live proctor or examinee is present and watch the live video of the students who are attempting online exams. Once appointment is made then students are taken to the proctoring room where proctor or professor is connected to the students via web cameras. Then student connect the computer screen with proctored screen. Now proctor can monitor the activities of the student by watching ion screen. Here few verifications took place that is proctor asked student to provide photo ID and few answer to the questions that will be the identity of the student. Now exam will start and here proctor can easily watch student screen. This way it is one of the secure solutions for the e-exam conduction and prevention of the cheating using monitored live screen and video. But the drawback of this type of proctoring that does not support continuous identification and second factor that made the solution not adoptable due to high internet bandwidth requirement. As the solution could only be adopted by developed countries with some other limitations. Few applications all developed under this category such as Proctor U [4], Examity [5] and software secure – PSI [6].
- 2) Recorded and Reviewed Proctoring:** Many research studies are available which have developed applications underlying in this category. These sessions are available as recorded videos because during exam session the computer monitors the students. Later, once exam is

completed then video can be reviewed by the human proctor any time. In this type of systems, the computer is by the student webcam is accessible to record the assessment. In this type of proctoring another factor is hardware cost is required to utilize best of camera feature. The student has to give permission to turn on the camera in entire session. Once exam is finished the recording is available to the instructor who can review details quickly and even watch the videos as long as he wants. Here, the limitation of this type of proctoring is as live proctoring. In addition to this this type of proctoring also called passive system. The logical limitation of this type of proctoring is to watch video to predict the behavior of the student and maintain the recorded sessions advantages. Unless no other solution is available here to detect the behavior of students. There are some commercial solutions available which perform the same action as described but with limitation of live proctoring. The market solution is available as Kryterion [1], ProctorExam [5], Respondus [9], Remote Proctor [2], ProctorCam [1], Virtual [11], and Learner verified [12].

3) Fully Automated Proctoring: There are few studies available, which proposed the complete end to end tools that cover both above categories. i.e., verification features such as biometric with other features question bank are utilized. For instance, a study Moukhliiss Ghizlane et al. [14] developed a complete software application which comprises the monitoring activities. Such types of proctoring is called passive system. In this system user must be active all of the time due to predefined texts and photo ID that captured continuously. One of the authentication technologies is recognition of the students through biometric or any physical image capturing based systems. These are typically built to verify these are the students who are registered for the exams. In these types of systems usually, facial recognition, voice recognition, fingerprints are used. In the last year, new biometric procedures are used such as keystrokes typing patterns that recognizes the speed, pressure, and style of the typing. Such type of verification is most popular when it is used with combination of another technology. Also, the monitoring activities are used in such type of systems such as webcam, microphone which replaced the live proctoring techniques and are presents in most of the online exam proctoring (OEP) systems. The camera is used to record the individual student is presenting in exams. Here, the camera is places to track the student or group of students to check the behavior weather they are performing cheating, receiving any kind of help from other student. The cheating could be happen using external hard devices such as mobile phones, books, webcam, microphone also required significant storage to store the video records and other data. Here, computer lock down can monitor the activity of the student

recorded in their computer which is carried out during the examination conduction that prevent the user from suffering through internet using any browser. In this type of system, the limitation is for high costly hardware requirements and high-speed internet requirement to store the high resolution videos.

Important Automated Online Exam Proctoring Techniques

There are various solutions introduced in pandemic (COVID19) that covers various characteristics of the Automated OEP. There are two types of solutions are introduced i.e., industrial based and commercial based. Those solutions which are proposed in literature are categorized as industrial solutions and those which are available in market to utilize are categorized commercial solutions.

Both type of solutions has some techniques which have their own features and also depend upon the need of the solution. Mostly machine learning based solutions are available which are very useful and easy to implement than other techniques. In section 3 the detailed analysis is performed on the tools and techniques. Here the importance of technique in Automated OEP is vary based upon the requirements. For example, for less development courtiers where financial factor is affected and to achieve the goal there must be available any solution that uses the technique less costly. Furthermore, there are other techniques found in literature and commercially available solution like NLP, genetic algorithm, CNN Based solution, rule based and fuzzy technique.

Also, different techniques are used based on the requirement of the solution. For example, research who carried out ML based approaches are using CNN algorithm to examine the verification of the users [16], cheating prevention [21][22] and for improving the abnormal behavior of the students during online exam [23][24][25]. Furthermore, that research who carried out the face recognition, head pose estimation, gestures identification, is mostly proposed own techniques [19][23]. Also, the NLP and genetic algorithms-based techniques are postponed in research are also analyzed in this section and included in given table. It is important to mention that some of the techniques are found where unique technique is proposed. For example, in study [48] the researcher carried out the formal method (FM) based approach in which used the quantified event automata to generate the OEP results.

1.2. Problem Statement

Currently most of the papers have used Automated OEP framework to target issues related to, biometric authentication, random question bank generation, keylogging, video recording, speech recording and various other monitoring activities of student's dishonesty during online examination. Research shows that to achieve the objective many machines learning-based approaches have been used in which NLP found most reliable with higher accuracy. It is analyzed from research that the OEP solutions are hard to adopt for less developed countries due to factors of costly hardware requirements, network dis connectivity issues etc. Also, all the commercial solution available are not public which made the adoption of OEP costly for the institutions in less developed countries such as Pakistan. Now all these issues have not been addressed before in research, Hence, open-source automated framework should be proposed to address offline monitoring that is cost effective. In addition, research shows that the commercially available solutions are not cost effective, and ease of use is less that require the training for both students and instructors.

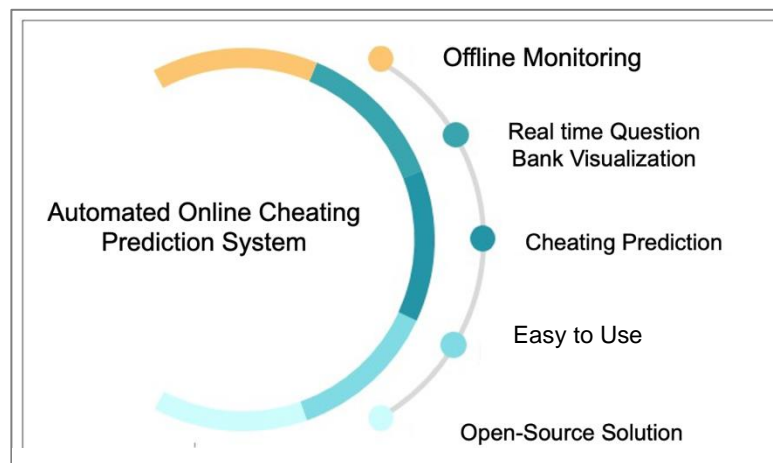


Figure 1- Problem Statement Summary

1.3. Proposed Methodology

Entire research is done in a very systematic way. **Figure 1.2** represents the flow of research step by step. In first step we identify the problem. Then proposed the ideal solution for the problem identified in first step. We carried out a detailed and comprehensive literature review which helps

us to identify the optimal solution for the problem. We reviewed the research carried out related to our proposed solution, analyzed and compared it. Then we implemented our framework using some tools and techniques. Our proposed framework is then validated using some 10 sample results from students.

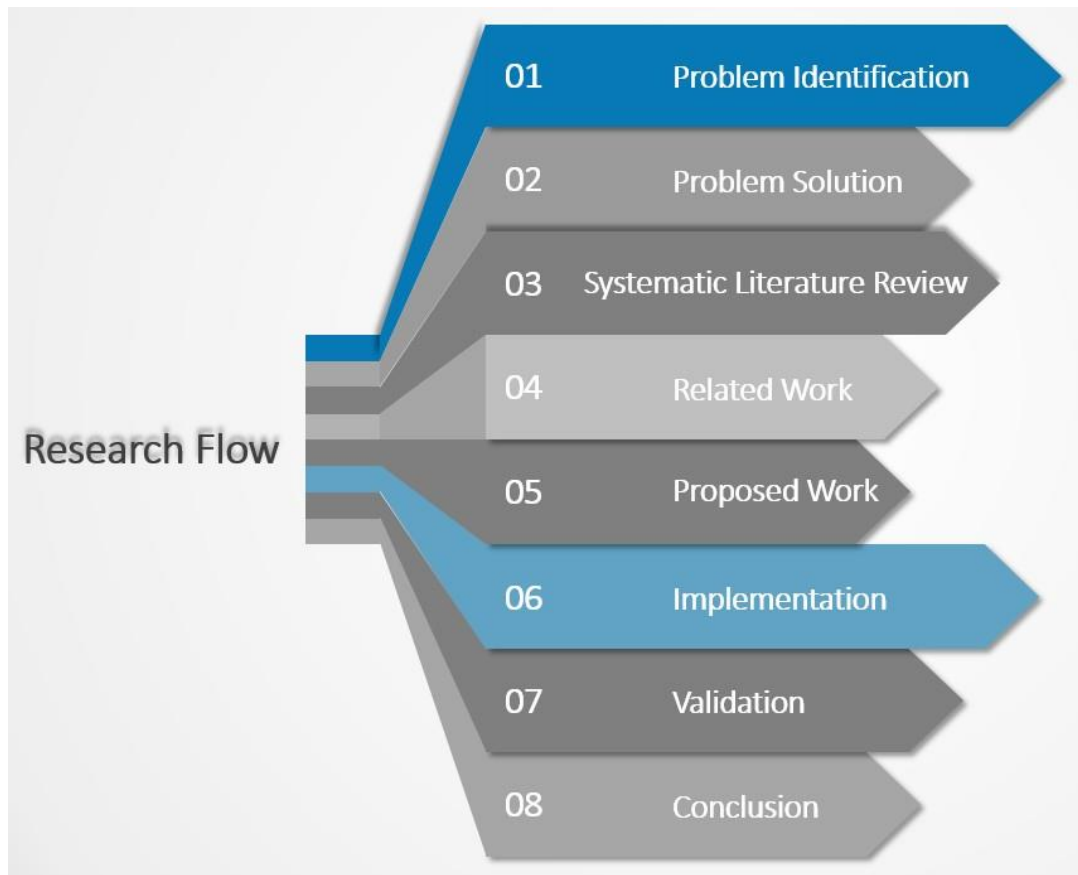


Figure 2- Research Flow

1.4. Research Contribution

The contribution of our research is a complete, open-source framework for Automated Online Cheating Detection for less developed countries. Detailed set of contributions of the proposed approach are as follows:

- Improve ease of usability of Software application in particular because of high training requirements of students and instructors.
- No hardware requirements to make cost effective solution for less developed countries.
- Minimal bandwidth network requirement to make use in offline monitoring of student.
- Real time question bank retrieval from cloud-based infrastructure.
- Easy to integrate with existing ERP or CMS system with partial automated report generation.

1.5. Thesis Organization

Organization of the thesis is represented in **Figure 1.3**. **CHAPTER 1:** offers a brief introduction containing the background study, problem statement, research contribution and thesis organization. **CHAPTER 2:** provides the detailed literature review highlighting the work done in the domain of Automated online cheating prediction solutions and more in general for online exam proctoring. Section one presents a systematic literature review on online exam proctoring techniques and tools. Section 2 describes the code analysis review from industrial perspective by presenting a review of all the different cheating prediction tools available in Market for various languages. **CHAPTER3:** covers the details of proposed methodology used for identification and solving of the problem in hand. **CHAPTER 4:** presents the detailed implementation of our framework, architecture along with its interface. **CHAPTER 5:** provides the validation performed for our proposed methodology using different data set of students. **CHAPTER 6:** contains a brief discussion on the work done and contains the limitations to our research. **CHAPTER 7:** concludes the research and recommends a future work for the research.

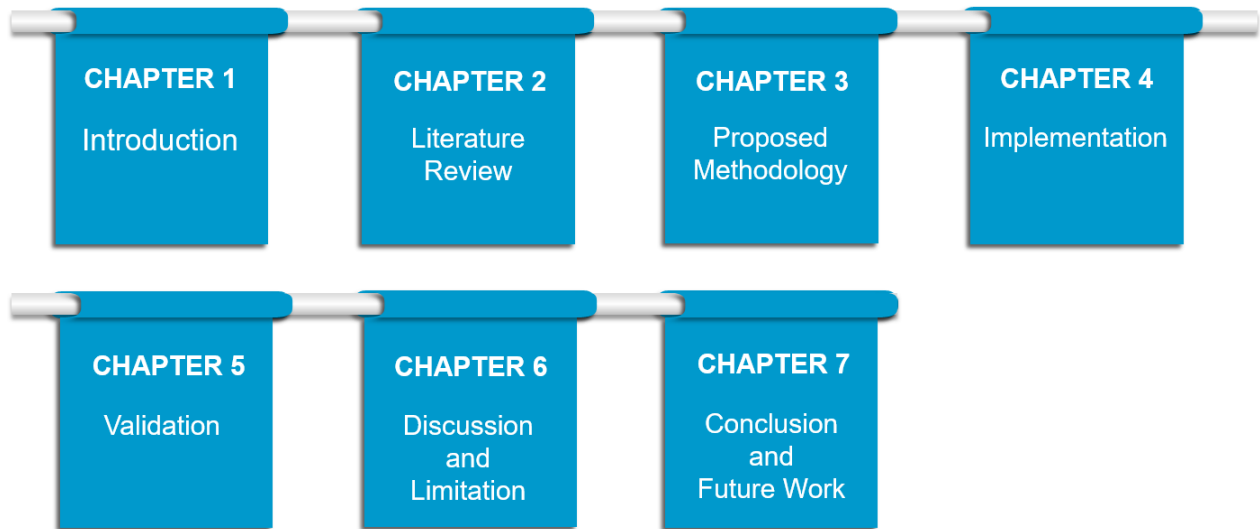


Figure 3 - Thesis Outline

Chapter 2

Literature Review

CHAPTER 2: LITERATURE REVIEW

In the past few decades E-learning in higher education is increased and play vital role in pandemic like COVID-19. Particularly, online examinations are conducted on e-learning platforms that leads to many security and cheating issues. For this reason, numerous research is available that proposes methodologies and techniques for seamless execution of online examination. However, it hard to find any study that provides the latest systematic literature review of anti-cheat or cheating prediction techniques and approaches in literature. This study provides the comparative analysis of online examination techniques, tools performed on 51 studies from last five years 2017 to 2021. Furthermore, in this time duration five most leading cheating prevention features are identified. The best frequent development approaches are also figured out in this literature review. Moreover, 14 important techniques which are mostly used in this time duration are found and 10 datasets including both public and private are identified. Proceeding towards the proposed solution, total 20 tools for the anti-cheat examinations are listed down. Almost 23 leading existing tools which are used in the literature are also highlighted. To narrow down the criteria for selection of best online anti-cheat examination solution adoption in different countries are also investigated. Finally, the overall cost of the e-learning infrastructure, specifically for conduction of examinations are determined by comparing the key factors of the global adoption with major online exam feature.

The scope of study is further restricted on the following:

- The different studies reported in the literature for anti-cheat examination particularly in E-learning.
- Tools proposed by researchers for anti-cheat examination in less developed countries.
- Offline monitoring feature with Cost-effective solution for less developed countries.

2.1. Systematic Literature Review

2.1.1. Review Protocol

To develop the review protocol that includes the various step as per guidelines. There are two components of the review protocol i.e., research questions and background of the study have been discussed in the last section i.e., introduction. This section presents the remaining four important components out of the total six components of the review protocol i.e., inclusion and exclusion criteria, search process quality assessment and data extraction/synthesis.

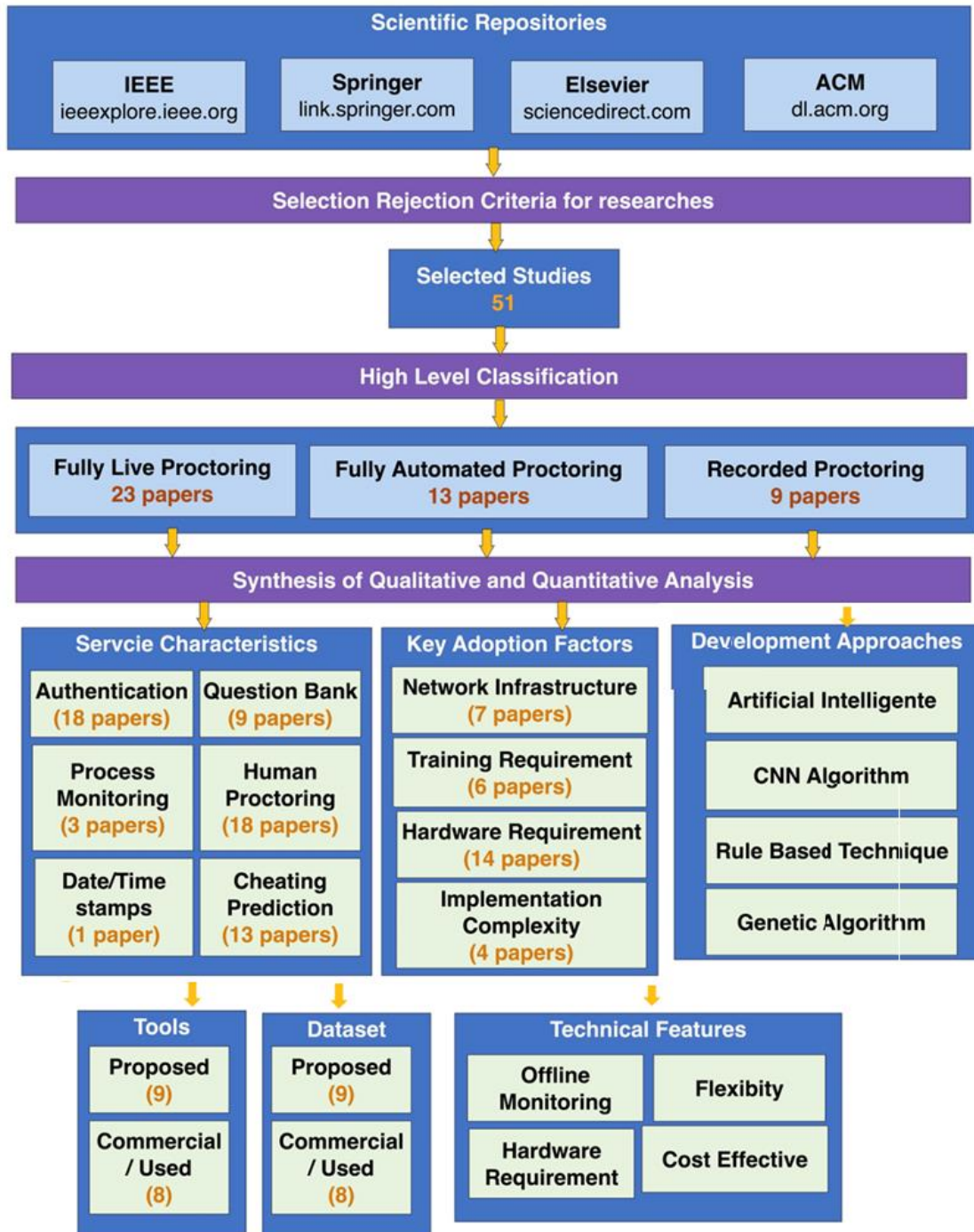


Figure 4- Overview of review process

A. Categories of Online Examination Proctoring

To simplify the data extraction and synthesis process the online examination proctoring is categorized into following categories.

1) Fully live Online Proctoring: Live proctoring is a type of service provide live proctor or examinee is present and watch the live video of the students who are attempting online exams. Once appointment is made then students are taken to the proctoring room where proctor or professor is connected to the students via web cameras. Then student connect the computer screen with proctored screen. Now proctor can monitor the activities of the student by watching ion screen. Here few verifications took place that is proctor asked student to provide photo ID and few answer to the questions that will be the identity of the student. Now exam will start and here proctor can easily watch student screen. This way it is one of the secure solutions for the e-exam conduction and prevention of the cheating using monitored live screen and video. But the drawback of this type of proctoring that does not support continuous identification and second factor that made the solution not adoptable due to high internet bandwidth requirement. As the solution could only be adopted by developed countries with some other limitations. Few applications all developed under this category such as Proctor U [4], Examity [5] and software secure – PSI [6].

2) Recorded and Reviewed Proctoring: Many research studies are available which have developed applications underlying in this category. These sessions are available as recorded videos because during exam session the computer monitors the students. Later, once exam is completed then video can be reviewed by the human proctored anytime. In this type of systems, the computer is by the student webcam is accessible to record the assessment. In this type of proctoring another factor is hardware cost is required to utilize best of camera feature. The student has to give permission to turn on the camera in entire session. Once exam is finished the recording is available to the instructor who can review details quickly and even watch the videos as long as he wants. Here, the limitation of this type of proctoring is as live proctoring. In addition to this this type of proctoring also called passive system. The logical limitation of this type of proctoring is to watch video to predict the behavior of the student and maintain the recorded sessions advantages. Unless no other solution is available here to detect the behavior of students. There are some commercial solutions available which perform the same action as

described but with limitation of live proctoring. The market solution is available as Kryterion [1], ProctorExam [5], Respondus 9], Remote Proctor [2], ProctorCam [1], Virtual [11], and Learner verified [12].

a) Fully Automated Proctoring: There are few studies available, which proposed the complete end to end tools that cover both above categories. i.e., verification features such as biometric with other features question bank are utilized. For instance, a study Moukhliiss Ghizlane et al. [14] developed a complete software application which comprises the monitoring activities. Such types of proctoring is called passive system. In this system user must be active all the time due to predefined texts and photo ID that captured continuously. One of the authentications technologies is recognition of the students through biometric or any physical image capturing based systems. These are typically built to verify these are the students who are registered for the exams. In these types of systems usually, facial recognition, voice recognition, fingerprints are used. In the last year, new biometric procedures are used such as keystrokes typing patterns that recognizes the speed, pressure, and style of the typing. Such type of verification is most popular when it is used with combination of another technology. Also, the monitoring activities are used in such type of systems such as webcam, microphone which replaced the live proctoring techniques and are presents in most of the online exam proctoring (OEP) systems. The camera is used to record the individual student is presenting in exams. Here, the camera is places to track the student or group of students to check the behavior weather they are performing cheating, receiving any kind of help from other student. The cheating could be happen using external her devices such as mobile phones, books, webcam, microphone also required significant storage to store the video records and other data.

Here, computer lock down are able to monitor the activity of the student recorded in their computer which I carried out during the examination conduction that prevent the user from suffering through internet using any browser. In this type of system, the limitation is for high costly hardware requirements and high-speed internet requirement to store the high-resolution videos.

A. Selection Rejection Criteria

To perform the comprehensive SLR, the inclusion and exclusion criteria is very impotent. For

this purpose, we have selected only those research studies which fulfill the below parameters:

- In the SLR the online examination is the major subject and only those research studies are selected which match the “Online Examination” aspects and other than this all studies are discarded. For example, one of the studies [64], in which a solution is proposed that cover the aspects of learning only.
- Only those studies are selected in which tools, framework and methodology is proposed fulfilling the anti-cheat online examinations.
- Only those publications are selected that are from 2017 to 2021 and other than these are rejected to ensure the SLR is performed on latest publications.
- The studies are selected from well-known publishers i.e., Springer, Elsevier, ACM, and IEEE. Other than these all studies are excluded.
- In case any study is found where content is quite similar then one of them is selected that suit best according to criteria.
- Those publications that are not fully addressed more than 2,3 pages are discarded.
- Only research studies are selected which are written in English language.
- Research papers that are perform the anti-cheat proposal only for online examination are included but other such as used in exam management are excluded.

We have performed the SLR based on inclusion and exclusion criteria. The studies discarded if any inclusion criteria is violated.

B. Search Process

we have performed the search process through major 4 data bases (IEEE, ACM, Elsevier, and springer) as mentioned in inclusion and exclusion section B. The graphical representation is shown below to summaries the search process. All the below explained review process fulfill the criteria of inclusion and exclusion. As the paper presents the advancement of the anti-cheat online examination therefore all the recent studies were selected.

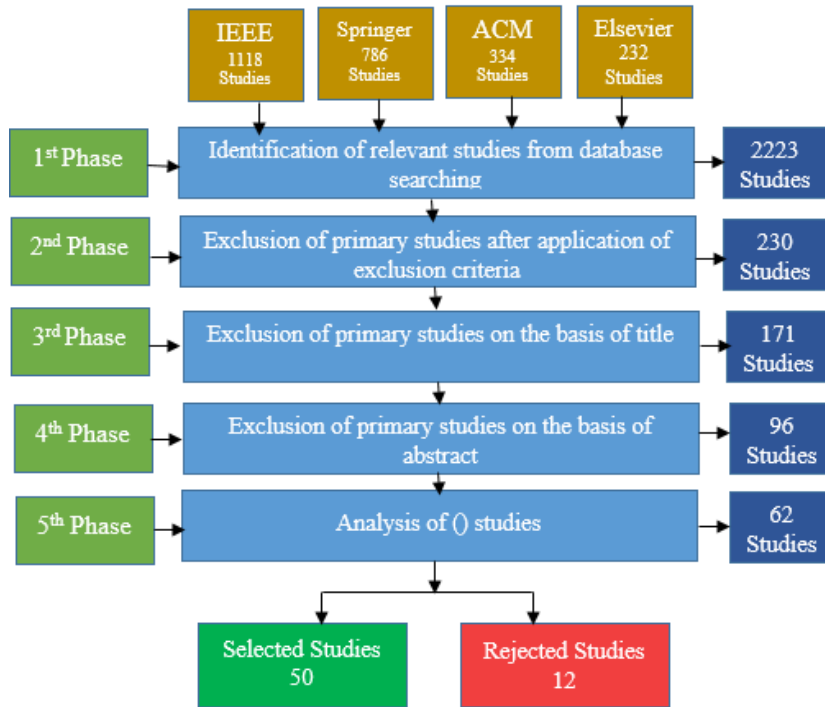


Figure 5- Search Process Flow

Furthermore, to search relevant studies, specific search terms are used as shown in table 1. To search throughout the different databases different keywords are used with combination of operators (AND, OR). By using the advance search feature such as time frame is imposed to select most relevant search results during 2017 to 2021.

Similarly, we have received 141 results against keyword “online exam” search term from IEEE database after applying above year filter i.e., (2017-2021). Also onwards, searching the keyword “e-learning exam” gave 48 results. Likewise different keywords have different results given in below table. In the next phase, we have analyzed total 1476 studies through search results. After applying different filters such as by checking the title of the studies, checking the abstract of the recaches publications are shortlisted to 469 studies. In the final phase we went through the general study of shortlisted 469 studies and search results are narrow down to 132 studies. The detailed analysis is performed on these publications to verify either these are compliance with inclusion or exclusion criteria. Also, few studies are found duplicated and rejected as mentioned in the inclusion and exclusion criteria. Finally total 51 studies are found which are fully compliant with the criteria we have finalized.

Table 1- Summary of Search Terms and Corresponding Results.

No	Search Term	Databases				
		Operator	IEEE	ACM	Springer	Elsevier
1	Online Exam, Biometric	AND	121	141	123	98
2	E-Learning Exams, Cheating Prediction	AND	42	35	16	53
3	Digital Exams	AND	57	49	44	43
4	Electronic Exams	AND	105	96	101	89
5	Exam Proctoring, Online Exam	AND/OR	9	5	14	18
6	Exam Authentication, Online Examination	AND/OR	254	147	197	134

7	Anti-Cheat software application, Online Exam,	AND/OR	19	11	13	9
8	Online Question Bank	AND	39	29	19	14

C. Quality Assessment

The inclusion and exclusion criteria is developed to support high quality results of the systematic literature review. Moving onwards to narrow down the search results and improving the quality additional parameter year is selected. The results are identified as current research on online examination. The year wise selection of studies is given in below table 2. The analysis of year wise studies identified as from total studies column i.e., total 7, 11, 13,12 and 8 studies found from the year 2017, 2018 and 2019 respectively. It reflects the total 84% studies are published during pandemic years i.e., 2017 to 2020 and only 16% studies are published in year 2021. hence, the findings selected from the studies are most recent.

Table 2- Summary of research papers based on publication year

Sr. No	Year	Studies	Total
1	2017	[13],[11],[6],[18],[29],[5],[12]	7
2	2018	[32],[23],[13],[17],[21],[26],[27],[23],[31],[26],[27],[23],[31]	11
3	2019	[12],[19],[121],[21],[38],[37],[1],[5],[6],[17],[13],[26],[27],[23],[31]	13
4	2020	[13],[17],[13],[17],[13],[17],[13],[26],[27],[23],[31],[26],[27],[23],[31]	12
5	2021	[13],[17],[13],[17],[13],[17],[13],[26],[27]	8

Now one of the most renowned parameters i.e., scientific database is considered in the selection and exclusion criteria which is shown in below table. The summary of the selected studies from the different databases are shown as total column with 22 studies from IEEE, 11 studies selected from Elsevier, 13 studies selected from ACM and 9 studies selected from springer. Here after detailed analysis of all these studies form different databases we found the most relevant studies from four popular databases i.e., IEEE and ACM.

Table 3- Summary of research papers based on scientific database

Sr. No	Databases	Conference Papers	Total
1	IEEE	[13],[11], [6],[18], [29],[5], [12]	7
2	Elsevier	[32],[23], [13],[17], [21],[26], [27], [23], [31], [26], [27], [23], [31],	11
3	ACM	[12],[19], [121],[21], [38],[37], [1], [5], [6], [17], [13], [26], [27]	13
5	Springer	[13], [17], [13],[17], [13], [26], [27]	8

Now to achieve the goal for more reliable results the search results are narrow down to the specific popular databases such as IEEE, ACM, springer, and Elsevier. Hence, total 31 studies are selected from IEEE, including 6 journal paper and 25 conference paper. Similarly, total 8 papers are collected from ACM in which 5 are conference papers other than that we have 3 journal papers from selected studies. Also, from springer database we have chosen total 6 paper which are all conference papers. Now represent in the tabular summary below table is shown with databases selection then category wise distribution of the studies that is either journal or conference paper and then total is shown at the last of the column.

Table 4- Summary of research papers based on scientific database and Journal & Conference Papers

Sr. No	Databases	Journal papers	Conference Papers	Total
1	IEEE	11], [6], [18]	[13],[11], [6],[18], [29],[5], [12]	7
2	Elsevier	[23], [31], [26], [27], [23],	[32],[23], [13],[17], [21],[26], [27], [23], [31], [26], [27], [23], [31],	11

3	ACM	[1], [5], [6]	[12],[19], [121],[21], [38],[37], [1], [5], [6], [17], [13], [26], [27]	13
5	Springer	17], [13]	[13], [17], [13],[17], [13], [26], [27]	8

B. Data Extraction and Synthesis

Once the inclusion and exclusion criteria is finalized and studies are selected then data extraction and synthesis process is executed. This process is very necessary that leads to the quality and required data for the research. Initially the primary elements of the data extraction are listed in table with names mentioned as bibliography information, proposed methodology, implementation details and outcomes are proposed as categorization of each study. After processing the primary data extraction and synthesis process other elements are selected as categorization of the studies, features of the online examination analyzed in each study, development approach which is used in each study, technique or algorithm used in each study. Furthermore, tools and datasets are also analyzed in each study. Finally, the comparative analysis of the major categories of recorded and reviewed proctoring, fully online proctoring, fully automated solution.

Table 5- Data Extraction & Synthesis

Sr. No	Description	Detail
1	Bibliography information	The title of the paper, publication year, type of publication such as journal, conference paper is observed of every selected studies.
2	Proposed Methodology	The methodology proposed in each research study is observed through.
3	Implementation Details	Techniques used in research study,
5	Outcomes Grouping	Outcomes, consequences of each study is analyzed The grouping is made for selected studies based on the categories and sub categories then results are

			summarized in below table.
6	Investigation of Categories / Characteristic / Technical Features		Here the analysis of classification of each major categories that is, fully live online proctoring, recorded and reviewed proctoring, fully automated solution. Here the RQ1 is discussed in section II A, B and C respectively. Also the analysis of sub categories such as, authentication technology, real time monitoring technology, Key logs, Active processes, question banking, usability, security, Human supervision, Internet Connectivity handling
7	Development approaches		Programming language is being analyzed in all studies which are presented in table.
8	Tools		All research studies are analyzed with targeted platforms used and presented in table.
9	Dataset		Target use is analyzed in all of the studies. Which is presented in table.
10	Techniques / Algorithm		The tools which are used in all of the studies are presented in table.
11	Key adoption factors		The summary of the standards to which analyze and which method is being used to check the compliance of is reviewed and presented in the table.
12	Comparative Analysis		The comparative analysis is performed of the major categories with respect to characteristics and features of categories presented in table.

2.1.2. Classification and Analysis

To answer the research questions mentioned before, a total number of research studies are found around 51, in which fully live online proctoring is examined out of which 15 journals and 34 are conferences. The figure presents the conferences and journals. Approximately there are 22% research studies which are published as journal and almost 78% are international conferences. There are studies which are published in different journals including, IEEE international journal of emerging and technology learn. Likewise, some other journal such as ACM and ScienceDirect

journals of wide range conferences are included in this study. Furthermore, all these selected research studies have been divided among three categories and further these are divided into sub categories as shown below.

Table 6- Synthesis Results - OEM Categories

Categories	Definition	Conference Papers	Total
Fully Live Proctoring	The real-time proctoring in which human is involved and behavioral analysis is performed.	[13],[11], [6],[18], [29],[5], [12]	23
Recorded and Reviewed Proctoring	It involves the video recordings and log details. Also human intervention is required.	[32],[23], [13],[17], [21],[26], [27], [23], [31], [26], [27], [23], [31]	13
Fully Automated Solution	More advance version where human proctoring is not required. system identified the fraud or cheating.	[12],[19], [121],[21], [38],[37], [1], [5], [6], [17], [13], [26], [27],[39], [40],[41],[44]	17

A. Fully Live Proctoring

Live proctoring is a type of service provide live proctor or examinee is present and watch the live video of the students who are attempting online exams. Once appointment is made then students are taken to the proctoring room where proctor or professor is connected to the students via web cameras. Then student connect the computer screen with proctored screen. Now proctor can monitor the activities of the student by watching ion screen. Here few verifications took place that is proctor asked student to provide photo ID and few answer to the questions that will be the identity of the student. Now exam will start and here proctor can easily watch student screen. This way it is one of the secure solutions for the e-exam conduction and prevention of the cheating using monitored live screen and video. But the drawback of this type of proctoring that does not support continuous identification and second factor that made the solution not adoptable due to high internet bandwidth requirement. As the solution could only be adopted by developed

countries with some other limitations. Few applications all developed under this category such as Proctor U [4], Examity [5] and software secure – PSI [6].

B. Recorded and Reviewed Proctoring

Many research studies are available which are have developed applications underlying in this category. These sessions are available as recorded videos because during exam session the computer monitors the students. Later on, once exam is completed then video can be reviewed by the human proctored anytime. In this type of systems, the computer is by the student webcam is accessible to record the assessment. In this type of proctoring another factor is hardware cost is required to utilize best of camera feature. The student have to give permission to turn on the camera in entire session. Once exam is finished the recording is available to the instructor who can review details quickly and even watch the videos as long as he wants. Here, the limitation of this type of proctoring is as live proctoring. In addition to this this type of proctoring also called passive system. The logical limitation of this type of proctoring is to watch video to predict the behavior of the student and maintain the recorded sessions advantages. Unless no other solution is available here to detect the behavior of students. There are some commercial solutions available which perform the same action as described but with limitation of live proctoring. The market solution is available as Kryterion [1], ProctorExam [5], Respondus 9], Remote Proctor [2], ProctorCam [1], Virtual [11], and Learner verified [12].

C. Fully Automated Solution:

There are few studies available, which proposed the complete end to end tools that cover both above categories. i.e., verification features such as biometric with other features question bank are utilized. For instance, a study Moukhliiss Ghizlane et al. [14] developed a complete software application which comprises the monitoring activities. Such types of proctoring is called passive system. In this system user must be active all of the time due to predefined texts and photo ID that captured continuously. One of the authentication technology is recognition of the students through biometric or any physical image capturing based systems. These are typically build to verify these are the students who are registered for the exams. In these types of systems usually, facial recognition, voice recognition, fingerprints are used. In the last year, new biometric procedures are used such as keystrokes typing patterns that recognizes the speed, pressure and style of the typing. Such type of verification is most popular when it is used with combination of another technology. Also the monitoring activities are used in such type of systems such as webcam, microphone which replaced the live proctoring techniques and are presents in most of the online exam

proctoring (OEP) systems. The camera is used to record the individual student is presenting in exams. Here, the camera is places to track the student or group of students to check the behavior weather they are performing cheating, receiving any kind of help from other student. The cheating could be happen using external hard devices such as mobile phones, books, webcam, microphone also required significant storage to store the video records and other data.

Here, computer lock down are able to monitor the activity of the student recorded in their computer which is carried out during the examination conduction that prevent the user from suffering through internet using any browser. In this type of system the limitation is for high costly hardware requirements and high speed internet requirement to store the high resolution videos.

2.1.3. Analysis Results

In this section results are presented to provide authentic answers to research questions. Particularly, analysis is performed w.r.t development approaches, tools, datasets, technique / algorithms and key adoption factors. Also comparative analysis is performed of major primary categories i.e. fully live proctoring, recorded or reviewed proctoring and fully automated proctoring. Here the results of analysis of each of the given aspects is discussed in detail in the sub sequent sections.

A. Techniques / Approaches

As primary categories are classified and further classified into different areas of features and general one but to answer the RQ3, it is important to identify the key development approaches in selected studies. To achieve this we have selected mostly used approaches and presented in table. This table can be beneficial to the researchers and who are targeting to develop online exam proctoring fully automated with best results using most accurate approach. Also this analysis also beneficial to those researchers who want to identify the trend in adoption of approaches have been used in last five years. According to our study most of the studies are using machine learning (ML) [2][3][6][7][9][10][11][14][16][17][23][26][29][31][32]. One major reason of ML adoption in most of the researches is the feature selection technique, classification algorithms utilization in particular approaches on online exam proctoring. Such studies are placed in this category. Now Artificial intelligence (AI) [23][25][27][28][29][30] is found highly useable at second number in most of the researches due to the overlapping theories. The techniques underlying in this category are, natural language processing (NLP), dynamic programming, genetic algorithms and many other types of algorithms are fall in this category. The third most widely used technique under specific circumstances and objective

formal methods are being used in research studies, the techniques fall under this category are, z-notation [12][17][18][24][23], time automata etc. Here most of the studies which are in the domain of ended system in the context of online exam proctoring are placed under this category of formal methods. Another category is traditional development, in which mixture of programming languages such as C# / .Net, PHP [6][7][8][10][11][13] etc. are used to develop and OEP solution based on web or desktop based applications. Here an important phenomenon to this category is not involved with other category of machine learning (ML) AI or FM techniques. The category is particularly concerned with traditional approaches. Another category is made that utilizes the mixture of different techniques to use the multiple feature of the solution such as machine learning (ML) with traditional is used in research [10][19][23][24][25][28][32]. Also, AI is used in combination with traditional algorithm in most of the studies [15][19][20]. The conceptual frameworks [24][29][32][34] are proposed and also utilizes other types of techniques such as ML, FM, and traditional development are comprised in this additional category. The summary of the development approaches in selected studies is presented in table. We have identified total no of 11 studies where proposal is with machine leaning (ML) and total of 9 studies using AI approaches . for instance, Nandini and Maheswari [34] proposed an methodology that is utiliozing the ML and evaluating the answers of the questions in OEP. Furthermore, feature extraction technique is utilized in combination with classification method of naïve bayes [17]. The study [26], where AI based approach is introduced by combining the technique of data mining i.e. fuzzy logic and question banks results are merged. On the other side, total no of 9 studies found [2][3][6][7][9][10][11][14][16] which utilizes the traditional development approaches, such as discusses above, C# (.Net), PHP, JAVA frameworks without utilizing of the AI/ML/FM techniques. The study [52], utilizes the secure communication platform or methodology in OEM where fog computing is being used. The authors who did not utilizes these techniques are provided with proof of concept using ASP.NET, C#. Similar, [37] utilizes the pHP programming language and share the mythology on the OEM. The summary is comprised in the form of given table.

Table 7- OEM Techniques & Approaches

Sr. No	Category	Reference	Total
-----------	----------	-----------	-------

1	Machine Learning	[9],[10],[13],[17],[21],[26],[27],[23],[24],[26], [27], [30], [31],[32].[35].[39]	16
2	Artificial Intelligence	[5],[11],[12],[14],[15],[16],[18],[19],[22]	8
3	Formal methods	[36],[29], [32],[34]	4
4	Traditional Approaches	[13], [17], [13],[17], [13], [26], [27]	9
5	Other / General	[10][19][23][24][25][28][32]	8

B. Machine Learning Techniques

In this section all of the selected studies are categorized bases on the used technique or algorithm in the research which have been proposed to achieve the research objective in OEP. To proceeding with these identification of techniques in selected studies a summary table is presented in which most frequent techniques are listed down with respective reference paper. Also this analysis is useful to those researchers who want to choose the best techniques using in the most of researches and their own research could be carried out in these researches. In section 2.1.3 – B we have comprised the results of approaches now based on those researches analysis is narrow down to most useful techniques. For example researches who carried out ML based approaches are using CNN algorithm to examine the verification of the users [16], cheating prevention [21][22] and for improving the abnormal behavior of the students during online exam [23][24][25]. Furthermore, those researches who carried out the face recognition, head pose estimation, gestures identification, is mostly proposed own techniques [19][23]. Also the NLP and genetic algorithms based techniques are postponed in research are also analyzed in this section and included in given table. It is important to mention that some of the techniques are found where unique technique is proposed. For example, in study [48] the researcher carried out the formal method (FM) based approach in which used the quantified event automata to generate the OEP results.

Addition lay, those techniques which are being used and not utilized by most of the researchers are not part of this analysis. For example, the study [12] developed the unique authentication method in which components of supervision is also merged along that. Although not enough information is provided by the researcher therefore it is not part of presented techniques/algorithms approaches.

Table 8- Machine Learning Techniques

Sr. No	Algorithm/ Technique	Reference	Total
1	Natural language processing (NLP)	[3],[5],[7],[8],[10],[11],[12], [14], [15] [17],[19],[20],[22],[23],[27],[30],[33],[35],[37]	19
2	CNN based technique	[5],[11],[12],[14],[15],[16], [18],[19],[22]	8
3	Genetic algorithm	[36],[29], [32],[34]	4
4	Rule based Technique	[13], [17], [13]	3
5	HW /SW Virtualization	[17], [13], [26]	3
6	K-Means Techniques	[38],[37]	2
7	Fuzzy clustering technique	[12]	1
8	Bayesian Network based technique	[33],[36]	2
9	Rule based interface	[21]	1

C. Tools

In this section tools which have been proposed in the selected search studies are included as part of analysis. However, it is important to highlight that both existing commercial solutions

and proposed solutions have been identified in this entire analysis. The usability of the research analysis will be beneficial for those research who carried out the enhancement in the domain of OEP and right tool must be selected to fulfill the core requirement of the objective. Hence, there are total 12 proposed tools and 15 commercial available tools found in the research and presented in the table. The attributes of tools analysis is performed as first tool name is listed in table then availability of the tool and respective reference is listed in the third column. Also different programming languages are used to develop the commercial available solutions or tools such as in study [11] [17][19][20][28]. Now identified proposed tools in the selected studies are available for the public and some are not available which is listed down in table. All the identified proposed tools are relevant to studies of verification and identification of abnormal behavior, security feature of OEP, question bank generation with multiple techniques and evaluation feature etc. There are few studies where proposed tools are missing and we have not included it in the part of analysis. In the study [19], author explained the proposed model and describe the each aspect of the methodology but the details about the implementation of interface, technology and platform was missing. Also there are studies where technique and approaches are available without development. In [21], author proposed the OEP using formal methods and utilized to achieve the results of integrity. Here, tools interface is not required therefore such studies are included in the part of analysis phase. On the other hand, only one tool is available with all of the required information of the tool with source code [18]. Despite of these all of the proposed tools are partial completed due to factors explained above. Due to this factor further evaluation of such tools is not possible. interestingly tools which are available with source code link or downloadable link are login link [38], [exam systems 15]. Similarly another tool available with web link [23], where some basic information of the tool is available relevant to OEP and relevant data set was available. Now, we have presented those tools which are available commercially are founded around 15 tools, in which some of the tools are only authenticating the identity of the student and few only providing the evaluation feature of abnormal behavior during online exam or cheating evaluation of the student. Also, few commercial tools are providing audio recording, video recording and entire solution is recorded bases. Some solution cover only exam activities and few are totally live proctored that is based on human and are not saleable or few of them are fully automated which are non reliable due to many factored of evaluation. Few solutions are non-scientific which provide the solution by combining the different functionalities. Due to few missing

functionality of continuous offline monitoring which requires the high speed internet and wont be useable in less developed countries. These factors are explained further in the section of key adoption factors. Below table presenting the both proposed and commercial solution with references.

Table 9- Automated OEM Tools

Sr. No	Tool Name	Availability	Reference Study
Proposed Tools			
1	Online Examination System	NA	[9]
2	e-Testing System	NA	[6]
3	Automatic Evaluation System	NA	[5]
4	Ville	NA	[8]
5	Simple and Dynamic Examination System	Public	[10]
6	MoLearn System	NA	[11]
7	Exam Wizard	NA	[16]
8	Online Item Exam System	NA	[12]
9	FLEXauth	NA	[13]
Commercial Tools			
10	ProctorU	Public	[13]
11	Examity	Public	[9]
12	PSI	Public	[6]
14	Proctor Exam	Public	[5]
15	Kryterion	Public	[8]
16	Remote Proctor	Public	[10]
17	Proctorcam	Public	[11]
18	B Virtual	Public	[16]

D. Dataset

In this section dataset is analyzed in all of the selected research studies which has its own importance for reliable validation and proposed techniques. We have found few studies with

open source datasets and some of the studies doesn't have the open access to dataset. In this way we have found around 9 datasets which are used and proposed in the selected studies for the validation purpose. The datasets names are given in below with mentioning of the purpose of the dataset and third column is shown with availability of the datasets. Here another important factor to describe is the characteristics of the datasets which include video, text and image files. Also the reference of the research is being mentioned in the last reference column. Its hereby mentioning that only 5 datasets are publicly available. Also only one datasets is newly created weather other datasets are used in multiple studies by different researchers. Some of the datasets were not available to download from the source. The study [16], carried out a research to validate various researches such as teacher assistant and company exam. Few authors have given the less information about the dataset such as download link is missing and few studies comprises the dataset where information is totally missing. Also different researchers have applied test scenarios for the validation purpose such as tidy [32] validated the proposed techniques or approach through various participation of the students with different iteration of exam conduction. Also study [39] comprises the survey of various teachers to validate the study.

Table 10- Automated Proposed Solution Dataset

Sr. No	Dataset Name	Format	Availability	Reference Study
1	Question Dataset	Text	NA	[9]
2	Student Dataset	Text / CSV files	NA	[6]
3	Online Exam Items	Text / CSV files	NA	[5]
4	Assitments	Text / CSV files	Public	[8]
5	Video Dataset	Videos	NA	[10]
6	OULAD	Text / CSV files	Public	[11]
7	AFLW	Images	Public	[16]
8	EMNIST	Images	Public	[12]
9	(OEP)	Audio / Video	NA	[13]

E. Key Factors for Online Examination Adoption

In this section we highlighted the key adoption factors in each research that have been analyzed with respect to features, underdevelopment approaches, techniques, algorithms in the subject of online exam proctoring. In this analysis we have found that the key factors categorization based on developed and developing countries are very important and play important role in finding the factors. According to IMF [65], there are particularly those developed countries where excellent financial statuses and economic situation is better than those developing countries which are struggling with economic disbalance and financial crisis. Eventually, the developed countries have more funds to adopt the stable infrastructure of OEP as compare to developing countries. Therefore, we have analyzed that only 16 studies are contributed to OEP from developed countries and adoption is more frequent in such countries. In the comparison, less developed countries have presented work or those studies are only theoretical based or have proposed the solutions in the study and don't have any practical tools available. In this manner the less developed countries are not in the situation to adopt the online examination using high costly tools available in the market cosmetically which have been available form the developed countries approaches. The solution which have been proposed by developed countries are found more closely to adopt for online examination due to stable infrastructure. The study [16] have proposed a fully automated OEP solution for prevention of cheating. As the part of the research, the Autor has developed a dataset and make the dataset publicly available for future research or carried out the research further by other researchers. In this propose tools, different biometric methods have been utilized with the help of eternal hardware devices and live or continuous monitoring have been implemented. The results of the tools found highly accurate and affective for the cheating prevention in OEP. Eventually this solution have been proposed in one of the developed country and can not be adopted in the developing countries for monitoring if the students, verification and validation of the student. As the result of this analysis such solution require high internet bandwidth and stable network infrastructure that is not available in the developing or less developed countries. From the analysis we can also through the light on the financial situation and existing e-learning examination is highly important for the adoption of OEP and due to various reasons or factor online examination couldn't be adopted globally. Below are the key adoption factors which have been identified during this analysis.

- a. Network Infrastructure:** The factor is described for the network infrastructure for those countries which have been highlighted in above section and for the factors having importance in the context of excellent internet bandwidth is available to the attendees of online examination on different location in a less developed countries. Ideally, up to 100 MBs internet is assumed good bandwidth and high speed internet to run through the successful conduction of the online examination. In contrast with less developed or under developing countries where internet infrastructure is not stable then adopting of OEP is difficult in context with this factor.
- b. Hardware / Software requirements:** This factor refer to the various external or software devices required based on the OEP tool used based on the financial conditions in a particular country. As, developed discussed in the above section, most of the research tools or proposed tools have been utilizing the large amount external hardware device such as for verification or validations through biometric devices, behavioral analysis using cameras etc. for execution of online examination. The hardware requirement is more costly factor due to higher procurement cost.
- c. Implementation Complexity:** This factor is involved with complex solution of OEP for the development phase. This factor is more likely to linked with higher cost because development cost is relative according to the complexity and size of the online examination solution. Particularly, in the selected studies these solutions are developed with the machine learning and artificial intelligence techniques and approaches. These techniques usually cost higher then other techniques. Another factor is the programming language that have the impact of the higher implementation cost.
- d. Training Requirement:** The training factor is referred to training requirements which is required for the execution of online exam for both the examinee and the evaluation instructor or invigilator. For example, those students who don't have IT background then it may be difficult for managing Ing or conducting the online examination easily which is directly proportional to increase the training requirement of the online exam system. On the other hand, invigilator may also have more requirements for the training if the complexity is more higher.

These few characteristics or factors mentioned above have more importance in adopting the online examination features when implementing the real time environment. As discussed above, all these factors are impacting the over all cost of the online examination

adoption which is important to consider for less developed countries that made the challenging to adopt the OEP. For this purpose in this research solution is proposed particularly for less developed countries where above all adoption factors are presented. Pakistan is one of the under developing countries where network infrastructure, financial resources, economy, digital educational is challenging to adopt but this solution is developed to solve major key adoption factors.

All the studies analyzed in this category are further analyzed based on the key characteristics and technical features such as 1-Authentication; 2-Human Proctoring; 3-Cheating detection 4-Process monitoring; 5- Continuous Monitoring 6- Reliable Results 7- Customized Real-time Question Banks, 8- Date & Time Stamps; 9- Auto exam Conduction , key adoption factors includes, 10- low cost network infrastructure; 11-Less Hardware Requirements; 12- Low training requirement and technical features includes: 13-Offline Monitoring; 14- Flexibility; 15- Scalable 16- External Hardware Requirement; 17- Low Internet Connection; 18 -Cost effective, 19- cloud-based are mentioned in below table.

Table 11- Automated OEP Comparison of Approaches

	Service Characteristics								Adoption Factors					Technical Feature					
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Fully Live Proctoring																			
Proctor	✓	✓	×	×	✓	×	✓	✓	×	×	×	×	×	✓	×	×	×	×	×
Examity	✓	✓	×	×	✓	×	×	✓	×	×	×	×	×	×	×	×	×	×	×
PSI	✓	✓	×	×	✓	×	×	✓	×	×	×	×	×	×	×	×	×	×	×
Recorded and reviewed Proctoring																			
Proctor Exam	✓	✓	✓	×	✓	×	✓	✓	×	×	×	×	×	✓	×	×	×	×	×
Kryterion	✓	✓	✓	×	✓	×	×	✓	×	×	×	×	×	×	×	×	✓	✓	×
Remote Proctor	✓	✓	×	×	✓	×	×	✓	×	×	×	×	×	×	×	×	×	×	×
Proctorcam	✓	✓	✓	×	✓	×	✓	✓	×	×	×	×	×	✓	×	✓	×	×	×
B Virtual	✓	✓	×	×	✓	×	×	✓	×	×	×	×	×	×	×	×	×	×	✓
Fully Automated Solution																			
ExamSoft	✓	×	×	×	✓	×	✓	✓	×	×	×	×	×	✓	×	×	×	✓	×
Proctorio	✓	✓	✓	×	✓	×	×	✓	×	×	×	×	×	×	×	×	×	×	×
Proctortrack	✓	✓	✓	×	✓	×	×	✓	×	×	×	×	×	×	×	✓	✓	×	✓
Comprobo	✓	×	✓	×	✓	×	✓	✓	×	×	×	×	×	✓	×	×	×	✓	✓
Sumadi	✓	✓	✓	×	✓	×	×	✓	×	×	×	×	×	×	×	×	×	×	×
ProctorFree	✓	×	✓	×	✓	×	×	✓	×	×	×	×	×	×	×	✓	✓	×	✓
HonorLock	✓	✓	✓	×	✓	×	✓	✓	×	×	×	×	×	✓	×	×	×	✓	×
Proposed Solution	✓	×	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	×	×	×

2.2. Research Gap

This section deals with the research gap in the proposed solution in industry or commercial automation domain. Online exam proctoring has been in the research studies since the COVID-19 pandemics. A lot of frameworks have been proposed which can be seen in research and the market. After a detailed analysis of literature on Automated Online Exam Proctoring, now we have identified proposed frameworks in the selected studies are available for the public and some are not available which is listed down in table 11. All the identified proposed tools are relevant to studies of verification and identification of abnormal behavior, security feature of OEP, question bank generation with multiple techniques and evaluation feature etc. Moreover, 14 important techniques which are mostly used in this time duration are found and 10 datasets including both public and private are identified.

However, it is evident from the analysis results that none of mentioned framework intended for the cheating prediction in under development countries with low bandwidth requirement of network. Furthermore, the offline monitoring during online examination is not conducted in any framework. These studies were further analyzed based on the standards that the frameworks are monitoring activities during offline. This help us identify another important gap in the literature i.e., none of the studies done so far has proposed a technique or framework for the offline monitoring under minimal bandwidth of network which is the need for the les development countries.

Analysis from commercial perspective has been presented in Section 2.2 and a summary of commercial tools for cheating prediction in different languages is presented in table 11. As per analysis results none of the tools predicts the cheating during offline or network is not available. These tools focused more on improving the maintainability of software rather than ease of use of the software. Similarly, none of the commercial tools check cheating prediction with respect to complexity of architecture, integration with existing software and network infrastructure.

The gap identified can be summarized in the following three major points:

- 1) Existing studies focus on automating online examination proctoring, but no study found which address on “Offline monitoring” feature particular for less developed countries.[1]
- 2) Tools available in literature or commercially hard to find for continuous monitoring using minimal internet bandwidth requirement.

- 3) No open-source tool support in literature and industry that provide cost effective integrate able solution with existing CMS or ERP.
- 4) In studies most of the frameworks require training but no study found which address the ease of use of application.
- 5) Hence, there is a need for open-source tool for the real time online examination cheating prediction that suits for under developing countries with ease, low network bandwidth requirement and must be cost effective.

Chapter 3

Proposed Methodology

CHAPTER 3: PROPOSED METHODOLOGY

As discussed, earlier anti-cheat online examination system can greatly mitigate the problems on online exam proctoring in less developing countries and provide cost effective solution. As detailed analysis is shown in above section for proposed and commercial solution which are indeed provide cheating prediction but due to high cost and other factors these solutions are not adoptable in less developing countries. Many tools provided great authentication methods which are respect to detecting the cheating behavior of the students. Also, many other algorithms have been proposed to enhance the accuracy and performance of the solution. Hereby, all the provided proposed and commercial solutions are not better option for under developing countries such as Pakistan. Hence, we have proposed and open-source tool that will predict the cheating behavior of the students when internet is not connected, or offline mentoring will be conducting with less complex network infrastructure.

The purpose of this chapter is to give a detailed description of the concepts used in the proposed solution. Sub section 3.1 presents our solution idea for the problem discussed in previous section. Sub-section 3.2 discusses our proposed system Workflow and onwards the proposed methodology is discussed in sub section. Then cheating prediction Similarity and rules are discussed in Sub-section 3.5 and 3.6 respectively.

3.1. Solution Idea

Our Solution idea is to create an anti-cheat framework online examination that:

- Ensure software usability
- Automate continuous offline monitoring
- Implement low bandwidth network requirements
- Easy to integrate with CMS or ERP Solutions
- Detect cheating prediction based on NLP
- Generate automated reports to cloud.

3.2. Proposed System Workflow

A workflow diagram of the proposed system is presented in Figure 6. The workflow is explained in the following major steps.

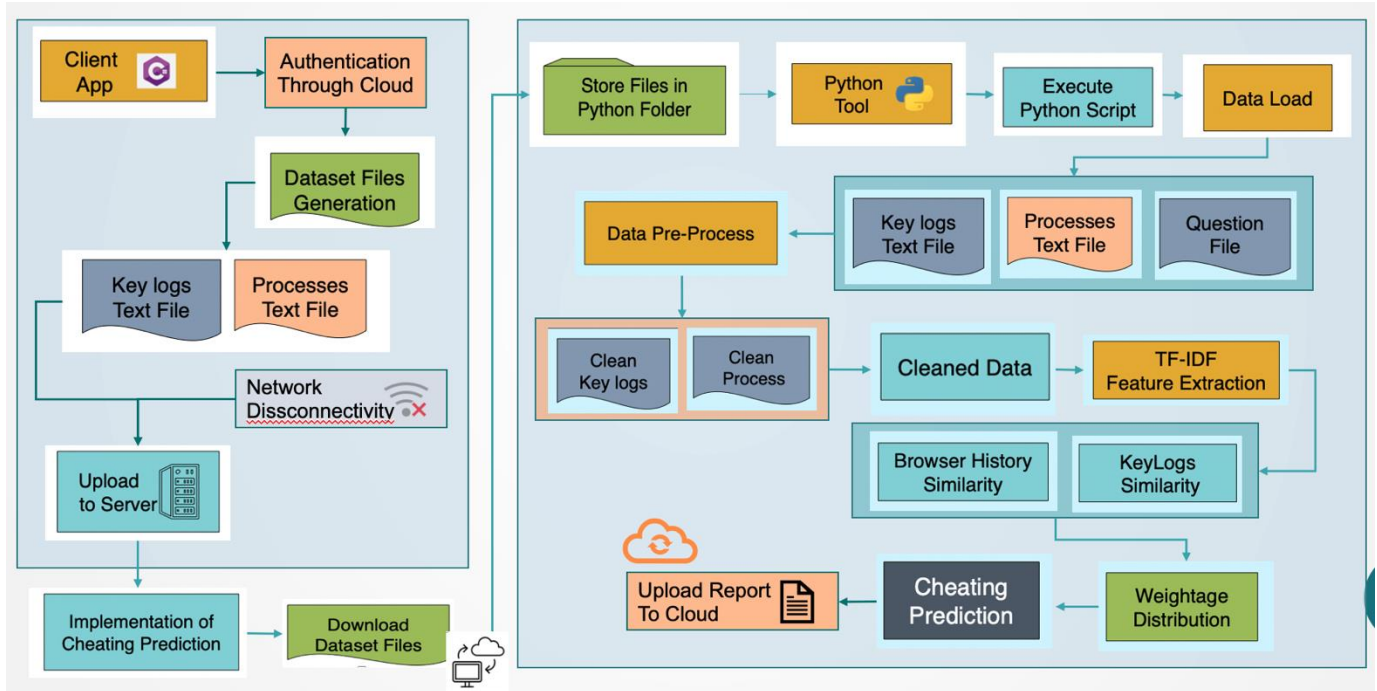


Figure 6- Proposed System Workflow

3.2.1. Cloud Based Architecture Specification

In this section Software architecture is explained which is necessary for any system to breakdown into component and how these components communicate with each other in effective manner that improve the efficiency of the model. In our proposed framework as shown below in Fig.1, we have used cloud-based architecture pattern which is distributed computing architecture that allocate the task between client and cloud server.

To automate the process for online examination conduction it is beneficial to remove dependency layer of integrating the desktop application with server preferences. To develop the stable desktop-based automation proposed framework uses cloud-based architecture pattern that suits best as explained below:

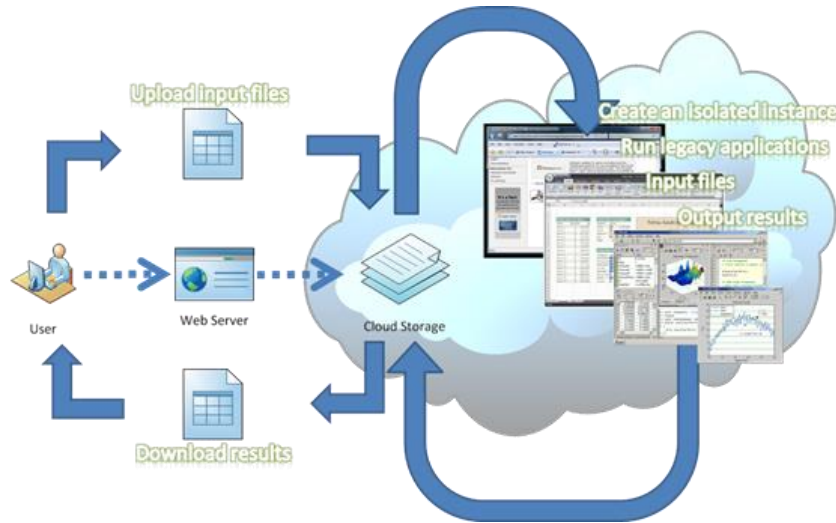


Figure 7-Cloud Based Architecture

3.2.2. Client / User layer System Functionality

In this section we will describe the client-side desktop-based application flow that is in AOCPS what the requirement are to build the desktop application which will be available to the users to install in their operating system and achieve the objective of this study. On client side only one user role is operating the application that is Student who supposed to appear in examination and finish the exam. Another role is come into picture that is Administrator to create the examination repository on cloud storage and monitor the cheating prediction of the students by generating the reports.

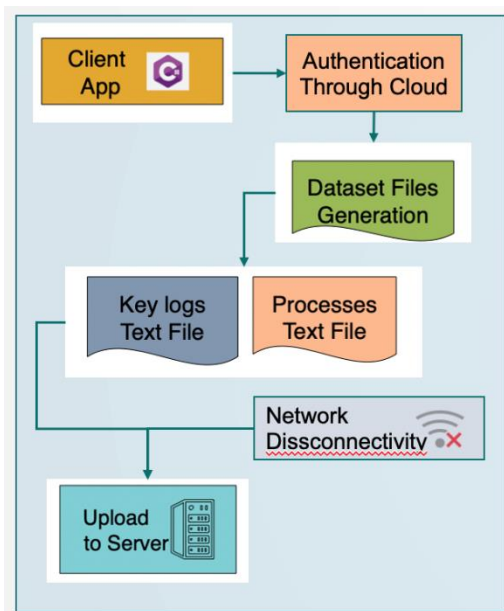


Figure 8-Client Application Flow

There are following modules available on client-side desktop application.

1. Desktop application source files on GIT (Open source & executable file)
2. Authentication
3. Visualization Realtime MCQs Question Bank
4. Keylogging
5. System Processes/ applications Monitoring
6. Browser History Monitoring
7. Mechanism for Dataset generation
8. Evaluation of Exam Conduction (Client-Side)

3.2.3. Desktop application Installation on Client Side

As the proposed framework is implemented by creating the desktop application for students who are supposed to appear in examination and attempt the test. To keep the application user friendly and easy to access for all users, an executable file is created which will be available on GIT repository to download and installed on any operating system by users. Once user has installed the application successfully next step to proceed with authentication.

3.2.4. Authentication

In this step, once user has installed the application successfully the login screen will be shown to authenticate the student. There are many authentication methods available which are used in literature but to achieve the objective of framework i.e., to conduct the cost effective and quick authentication student ids and passwords are pre generated by administration and provided to student via emails.

User Authentication

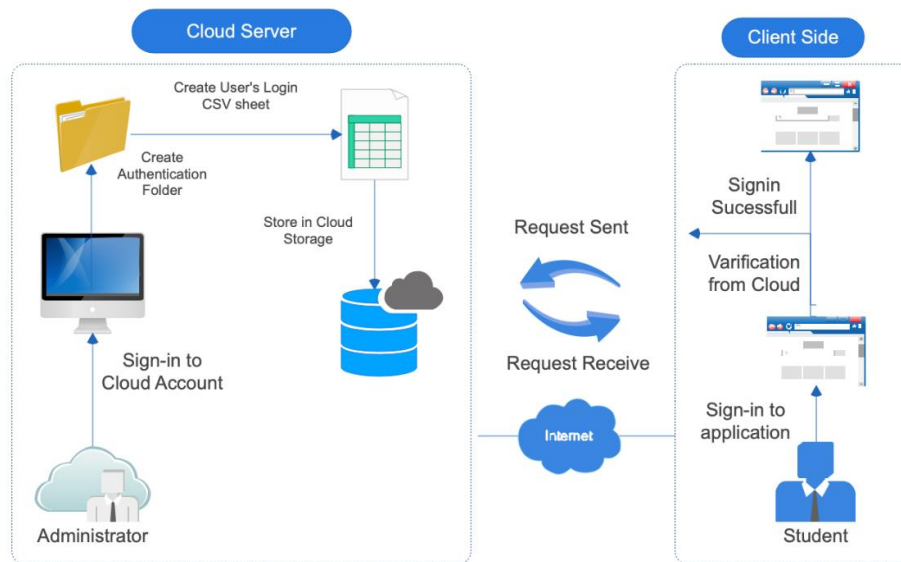


Figure 9-User Authentication Flow

3.2.5. Visualization of Realtime Question Bank

In this step, when user has successfully logged in to the application, an interface will be shown to attempt the exam. Here all the MCQs based questions was created by administrator and available on cloud storage. Here some validations are applied in designing the framework such as in case examination time exceeds the allotted time then a mechanism should be available to submit the examination in given time frame. For this purpose, timer function is used which is available from C# library, which will be enabled to monitor the execution time of the exam.

Question Bank Generation Architecture

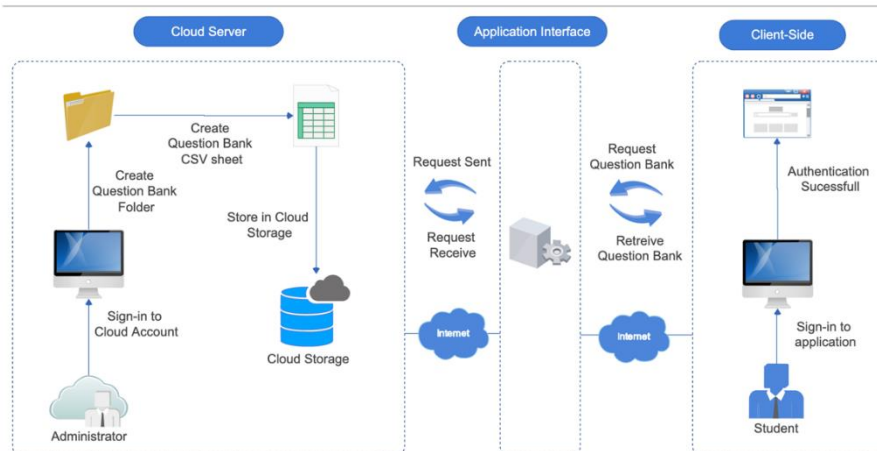


Figure 10-Question Bank Generation Flow

3.2.6. Keylogging

The objective of the proposed automated cheating prevention framework is based on this feature. The automated framework is designed in such a way to detect the user keystrokes hiddenly as soon

Key logs Generation Flow

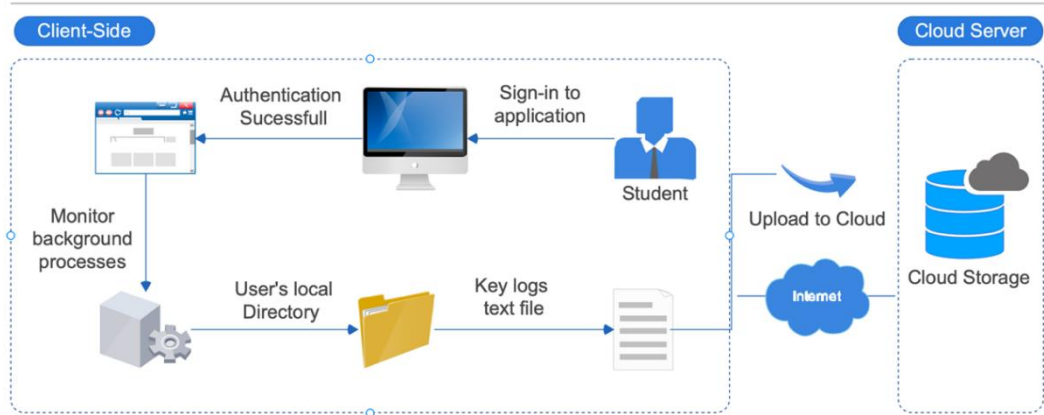


Figure 11- Key Logging Flow

as user start the examination, tracking gets started. The approach is designed with multiple perspective such as student should not be aware of tracking or monitoring activity on the operating system while attempting the question paper.

Entire user's keyboard is captured while attempting the question paper. The approach is designed for below concerns:

- What if user type invalid or special characters which are not useful in predicting the cheating of the student?
- Does key logs captured of only current date or when examination is being conducted?
- How does the key logs only captured of certain time frame while student is attempting the paper?
- How record of real time key logs generated and saved for future evaluation?

To ensure all the above concerns, the framework is designed to mitigate the chances of inaccurate and incomplete data should not be collected. Also, the approach is implemented to utilize the maximum prediction algorithm performance.

3.2.7. System Application History Monitoring

As discussed in above section user's keystrokes are logged hiddenly and stored in a local user's directory, likewise user's applications which are being used during the examination attempt or tracked hiddenly.

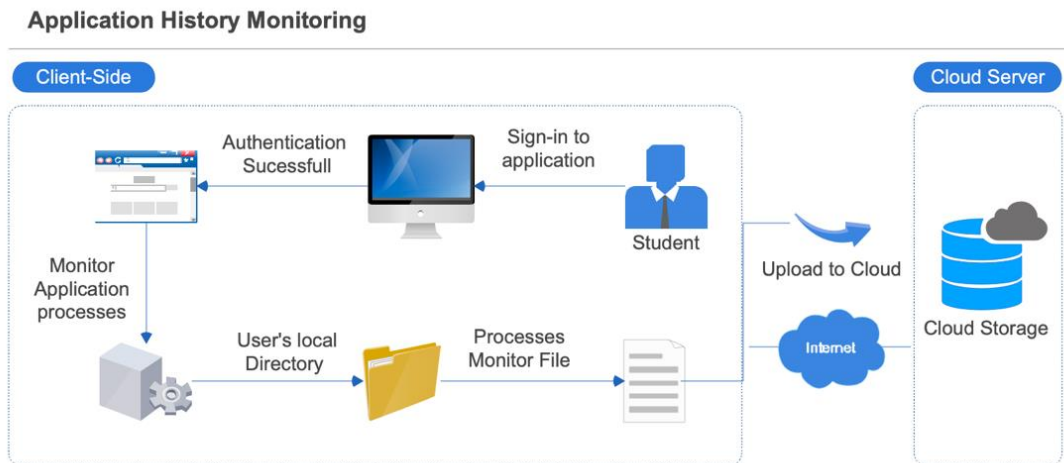


Figure 12- Application History Monitoring

The monitoring of application is also play important role in predicting the cheating probability during examination. There is many software available that prevent the user to open the browsers, restriction on closing or minimizing the window, but the proposed framework is designed to let user use all the application such as, browsers, communication or chatting desktop application, Skype, Microsoft office, WhatsApp, massager apps and many others. Another concern raised about the background processes running on user's system

The following concerns are facilitated while designing the approach for application history monitoring.

- The tracking of the application will only be tracked while user is attempting the question paper. As soon as user completed the examination and submit the paper successfully the application tracking will be logged off.
- In case, user has already opened many applications which are mentioned in below section then function is designed that trim out old application processes which are being used in background processes on user's system. only count current date and time applications on which examination is being conducted.
- By implementing the above rule, the dataset is more precise and increases the performance of the prediction algorithm.
- In case, user open and close the application multiple time then code is designed to track it properly and maintain the logs, for this purpose a logging loop cycle run after every one minute to refresh the application history and add or replace the concurrent event and logged into application history.
- The monitoring time of all the processes is also logged that help in predictive analysis and achieve the aim of the proposed solution.

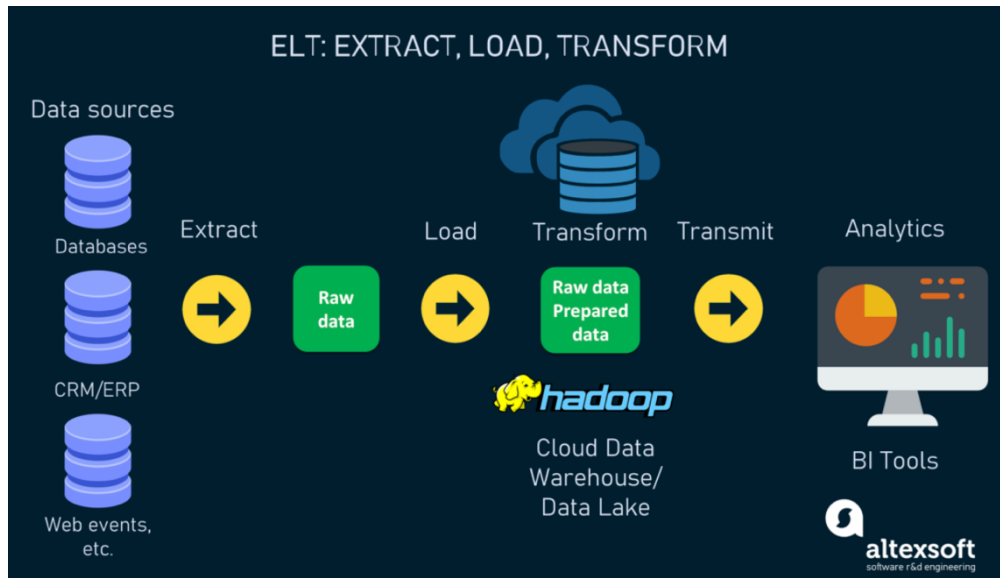


Figure 13- ETL Data Generation Approach

To cutdown the number of applications and browsers, the approach is being used by selecting specific number of browsers and applications which are listed below.

Also, the approach is designed in such a way that above rule which is selection of specific number of applications and browsers could be scaled up in future.

- Specific browser includes, chrome, Safari, Firefox, Internet explorer, Opera and Microsoft edge.
- Communication or chatting application includes, Skype, WhatsApp, Microsoft PowerPoint, PDFs, MS word, MS excel.

3.2.8. Mechanism for Dataset Generation

The crucial part to train any algorithm is to provide useful data on which some sets of procedures are implemented and achieve the objective of the proposed framework. Now in this section the mechanism will be discussed to convert gathered user's data into a useful dataset.

As all the user's key logs, processes, browsers and applications are being monitored and must be stored somewhere to keep the record of these for future processing. For this purpose, below mechanism followed up for dataset generation:

ELT Approach: The ELT stands for extract, load, and transformation. The data flow of ELT relies on three steps.

The first step is the extraction of data from pool of various sources, in our research the source of data is

user's real time key logs, application or browser history. Here full extraction method is used i.e., all key logs records and being extracted and saved in a user's system local directory where application is installed and being used. The purpose to use full extraction method is to ensure all data is being captured and can be filtered in later stages of transformation. Here raw and unstructured data is saved in form text files. The data is unstructured and categorized in two different files. All the user's key logs are saved in one text file and all the other applications and browser's history is saved in different files. Eventually two different text files are extracted and saved in user's directory which will be used in further processing.

Now in second step of ELT approach, extracted data files will be loaded up to a central storage from where it can be accessible and transformed for further processing. As discussed in above section cloud server architecture is used and here already stored data at user's local directory will be loaded on cloud server. The purpose of the ELT approach is basically utilizing the loading of raw data directly to the target cloud location which is more efficient process. The application is being utilized by the multiple user's folder structure is created while loading the extracted data files on cloud server with each user's name who is attempting the examination. The purpose to generate the folder structure with each user's name is to categorized data files which can be utilized in further transformation and algorithm that helps to generate the final cheating prediction against each user.

The third step of ELT is data transformation. The data transformation is the process of different activities aiming to prepare the data that fit's to required parameter of another system. Here transformation is taking place by third party python libraries that converts the data into arrays to use it for further processing. The data processing or cleaning is explained in section 2.3.

Data quality: During monitoring and logging the data into files few quality matrices are ensured. Here one key factor to manage the redundancy of the key logs by keeping the date and time stamp in record while logging. The date and time stamp are logged whenever user attempt the examination.

This timestamp is recorded because to ensure data fragment collected in that timeframe when the examination is conducted. This additional validation ensures the predictive analysis is more accurate to the respective user's activity performed during the examination. As discussed in section 2.2.3, the monitoring activity starts as soon as user start the examination and ended up when examination is submitted. During this time frame a timestamp also benefit when user attempt any other examination in future and the previous user's record will be available along old date and time stamp.

The application and browser's history monitoring are also logging with a date and time stamp that helps in collecting the particular data fragment collection during predictive analysis. The redundancy of the dataset is also managed by removing the duplicate processes. The mechanism of removing the same processes is handled by updating the logged time. This iteration of updating all applications and

browser's history by running iterative cycle after every one minute, explained in above section 2.2.5. In this way the redundancy of the data is mitigated, and size of test data reduced which enhanced the predictive analysis performance.

3.3. Cloud Sever System Functionality

3.3.1. Data cleaning and Pre-processing

Firstly, the available data gathered is not clean and may have invalid characters or errors that affect the overall productivity for the highest information in decision making. For this purpose, data cleaning through CSV readers due to raw dataset which are available in Text form and cleaning process applied on multiple categories of dataset i.e., Question bank, key logs and browser history or applications dataset.

Firstly, data cleaning process is applied on key logs data. As entire keyboard is captured during the examination conduction so that any information could not be skipped. The capturing took place keeping the QWERTY keyboard as a base keyboard and set rules accordingly to clean data. As only meaningful data is required from key logs data file, for this purpose special characters are removed by defining these in a list of arrays.

```
rep = { 'Space' : " " , 'Escape': "" , 'F2': "" , 'F3': "" , 'F4': "" , 'Delete': "" ,
'PrintScreen': "" ,
      'F5': "" , 'F6': "" , 'F7': "" , 'F8': "" , 'F9': "" , 'F10': "" , 'F11': "" , 'F12':
"", 'Pause': "" ,
      'Oemtilde': "" , 'OemQuestion': "" , 'RShiftKey': "" , 'LControlKey': "" , 'LWin': "" ,
'LWin': "" ,
      'Apps': "" , 'RControlKey': "" , 'Left': "" , 'Down': "" , 'Up': "" , 'Right': "" , 'Tab': " " ,
'Return': "" ,
      'LShiftKey': "" , 'Oemcomma': "" , 'OemPeriod': "" , 'Back': "" , 'OemOpenBrackets': "" ,
'Capital': "" }
```

Figure 14- Data Cleaning and Pre-processing

The above list of special characters will be removed from key logs. Now the raw data also included some extra spaces and all types of brackets then these are also cleaned from key logs data file.

```
KeyboardLines = re.sub(r'\[[^]]*\]', "\n", KeyboardLines)
```

```
KeyboardLines= KeyboardLines.replace("( )+", " ")
```

Now, all special characters, spaces and brackets are cleaned now convert the list of data file into separate lines to pick all meaningful words line by line during processing of the data.

```
lines = KeyboardLines.split("\n")
Key_lines = [line for line in lines if line.strip() != ""]
```

In above entire key logs data is cleaned and can be processed in further steps. Now browser and application history will be cleaned. Here all the special characters are also removed and split the data into browser's array and another array for applications.

3.3.2. Set Rules for Browser and Applications

In this step, cleaned and pre-processed data will be executed with different rules which will help in enhancing the accuracy of result then stored them in further processing. As there are hundreds of application processes and list of browser's data is available, but we filtered only selected data which can server with the better prediction of the proposed model. Following list of browsers and applications are shortlisted which are being utilized in further processing.

```
AppHis = ['WhatsApp', 'WINWORD', 'notepad', 'Skype']
Browser = ['firefox', 'chrome', 'explorer', 'opera', 'edge', 'pale Moon', 'safari'] |
```

For this purpose, few sets of browsers are defined in an array to filter only those data which are coming from pre-defined browsers for example, Chrome, Firefox, Microsoft edge, safari and opera are listed down. On other side few applications such as WhatsApp, Microsoft word, skype, notepad and PowerPoint are listed to store data in a separate array from these applications.

```
if(i.find('whatsapp')):
    whatsappcount = 1
if(i.find('Skype')):
    Skypecount = 1
if (i.find('notepad')):
    notepadcount = 1
if (i.find('WINWORD')):
    WINWORDcount = 1
```

Now mechanism is defined to set flag on all the applications, i.e., if found any application from above array then put FlagCount = 1, this indicates the application is being used while monitoring real-time processes as shown above. Likewise, only selected browser's processes will be fetched out from the data file and removed extra spaces from list and store in a list for further processing.

3.3.3. Set Rules to Calculate Weight / Time for Browser

Now, once cleaned, pre-processed data is filtered out according to above rules then new rule will be used to calculate the total time of the processes being logged during the monitoring of the processes. For this purpose, firstly, only current date data, which is logged during the examination will be captured, and old history will be truncated. Here, time stamp is logged in a format

“Day/Hour/Minutes/Seconds/Milliseconds”. As the day attribute will be truncated to apply the prediction of the single test and rest of the time will be utilized.

```
Brows_res = [sum(x) / len(x) for x in zip(*Browsing_time)]
del Brows_res[0] # remove Days i-e it is already within a Day
hours = (Brows_res[0]) * 3600
minutes = (Brows_res[1]) * 60
seconds = (Brows_res[2]) + (Brows_res[3] /1000)%60 # convert milliseconds into seconds
Brows_weight = hours + minutes + seconds
```

The rule specifically applied to mitigate the chances of errors in decision making. The less focused purpose of this rule to increase the radiality of the proposed system so that authentic results are generated.

4. Proposed Prediction Algorithm

All the Prediction model uses historical data to predict the future event and suggest the optimal actions to take. For this purpose, the cleaned and pre-processed data is passed through multiple set of rules that help in making the data useable for prediction model.

There are multiple techniques available to train the model, but our prediction model used TFIDF technique that gives better accuracy measures which is explained in below section.

4.2. Feature Extraction Technique

As feature extraction is an important part of training any model and to achieve higher accuracy. There are number of feature techniques are available, but TF-IDF is used that gives more accurate results on our pre-processed data. As two different statistical methods are used first term is TF (Term Frequency) that gives the total number of times a term appear in the document in contrast with total number of all words present in the document. In comparison IDF (Inverse Document Frequency) help to measure that weight of the given words in document and give which are common and rare number of words are used in entire document.

Before processing the data with TF-IDF, stop words are removed to lower the dimensional space of both documents, key logs and application or browser history. The built in NLTK corpus library is used to remove soapwords which comparatively more efficient and effective than other libraries.

As TFIDF is the dot product of two vectors and here the algorithm is designed with two different combinations are used, one dot product or TFIDF is calculated with combination of Question bank and key logs and other is Question bank with Browser history. The purpose of the different combinations is ultimately increasing the accuracy and precision measure of the prediction model by getting different similarity measures as explained below.

4.3. TF-IDF for Question Bank and Key logs Data

As the TFIDF works with dot product of two vectors here one vector is utilized as list of questions and second vector for key logs list to find the similarity. First both questions and key logs data is

combined and put it in an array. As the key log data is already cleaned in pre-processing step but to make the visibility of results after combining the data, special characters are re checked in document and removed along stop words. Below is the mechanism followed to find TFIDF for question and key logs data.

```
KeyDocument = pd.concat([Key_data, Question_data])
# removing special characters and stop words from the Keyboard Data
KeyDocument['documents_cleaned']=KeyDocument.Data.apply(lambda x: " ".join(re.sub(r'^a-zA-Z',' ',w).lower() for w in x.split() if re.sub(r'^a-zA-Z',' ',w).lower() not in stop_words_1) )
```

Now to find the key similarity of the above vectors we have used the dot product of above vectors as shown below.

```
Key_similarities=np.dot(tfidf_vectors_key,tfidf_vectors_key.T).toarray()
```

4.4. TF-IDF for Question Bank and Browser Data

In this step we will again combine the data of questions and browser's history that specifically searched by user on specific browsers and put it in an array. As the browser's history is already cleaned in pre-processing step but to make the visibility of results after combining the data, special characters are re-checked in document and removed along stop words. Below is the mechanism followed to find TFIDF for question and browser's data.

```
SearchDocument =pd.concat([Suspected_data, Question_data])
# removing special characters and stop words from the Searched Data from different Apps "
SearchDocument['documents_cleaned']=SearchDocument.Data.apply(lambda x: " ".join(re.sub(r'^a-zA-Z',' ',w).lower() for w in x.split() if re.sub(r'^a-zA-Z',' ',w).lower() not in stop_words_1) )
```

Now to find the key similarity of the above vectors we have used the dot product of above vectors as shown below.

```
Search_similarities=np.dot(tfidf_vectors_ques,tfidf_vectors_ques.T).toarray()
```

4.5. Average Similarity of Key Logs with Questions

In above steps the TFIDF is calculated with two different combinations and now the same results will be utilized to calculate the average similarity of the results with respect to question. This result will be processed in coming steps for cheating prediction.

Here firstly, we have calculated average similarity of key logs with respect to all questions then, we have calculated average similarity of browser's history with respect to each question as shown below:

```
for i in Search_similarities[:-30]:
    searchsum = searchsum + (i[:-30:])
    Totalsearch = Totalsearch+1
Avg_Search_similarity = searchsum/Totalsearch|
```

4.6. Converting Average Similarities to Probabilities

All the results calculated above are stored in list as an average result that needs to be convert in probability that normalize the values and give results. The input of values is taken in form of vectors which are stored in list above then convert them in vector of probability with given formula:

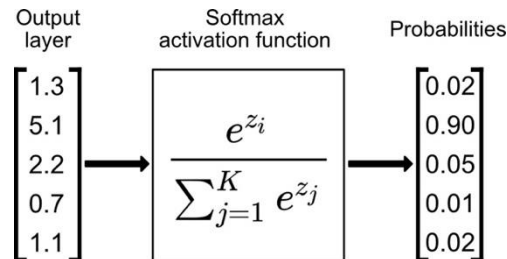


Figure 15- Formula for Probabilities

As the above formula uses the vector of some input data as we have in form or average similarities of keys and browser’s history with respect to questions then this result will be passed from SoftMax function imported from python library.

```
Search_result = softmax(Avg_Search_similarity)
Key_result = softmax(Avg_key_similarity)
```

As above function will store the probability vector is two variables “Search_results” and “Key_results” and then total prediction can be calculated from these two variables.

```
for i in range(30):
    index = i+1
    SearchResult.append([index,Question_data.iloc[i,][0] , Search_result[i]])
for i in range(30):
    KeyResult.append(Key_result[i])
```

4.7. Average Cheating Prediction Rule

In this step all the results we get from TFIDF, average similarities and probability results will be utilized by given percentage. As of now the cheating prediction rule is required to give weightage to the results because we have multiple sets of results with key logs, browser history and applications.

The rule is defined to give weightage to all of data as below:

Table 12- Cheating Prediction Weightage Distribution

Similarity Elements	Weightage%	Total
Key logs & Browser History Similarity (80%)		
Key logs Similarity	40%	40%
Browser History Similarity	60%	60%
Key logs Similarity + Browser History Similarity	40+60%	80%
Apps History (20%)		
Apps (WhatsApp/ Notepad / Skype / MS Word)	20%	20%
Key logs & Browser History Similarity (80%) + Apps History (20%)	80% + 20%	100%

Now browser's history and key logs are multiplied with percentage as defined in the rule and save the results in variables.

For application's data below formula is applied and save the results in variable. Here WhatsApp flag is given double weightage and multiplies with the value 2.

Now, to combine the above results below formula is applied and total cheating prediction is calculated.

Chapter 4

Implementation

CHAPTER 4: IMPLEMENTATION

This chapter provides the implementation details of our proposed framework. Section 4.1 This chapter provides the implementation details of proposed framework. Section 4.1 describes the architecture of our framework. Section 4.2 describe the description of the .Net technology. How cheating prediction algorithm is used described in Section 4.3. Finally, the client and administrator tool interface along with description is presented in Section 4.4 & 4.5. Our tool is open source and can be found here [1].

4.1. Automated Online Cheating Prevention System

Tool support is an important factor to increase the productivity of software development. A tool support architecture to support the framework is shown in Figure 4.1. The architecture comprises of two main components i.e., Client Side and Server Side. On Client-side visual studio framework and its subcomponents and utilities. The other one is server -side implementation tools which is anaconda also having different sub modules. .Net framework is used that provide the visual designer-based window forms to build the front end of desktop application.

Here multiple libraries such as cloudinary and Excel reader libraries are used in C# windows form application. There are also multiple libraries are available are used with in the .net framework to generate the user interface (UI) also to generate the dataset text files. The anaconda python compromises of multiple libraries which are used to build the cheating prediction algorithm. The dataset files generated by the C# window form application are used by the anaconda python and processed under different libraries mentioned in below diagram. To generate the results cloudinary APIs are used and explained in detail in Section 4.5.

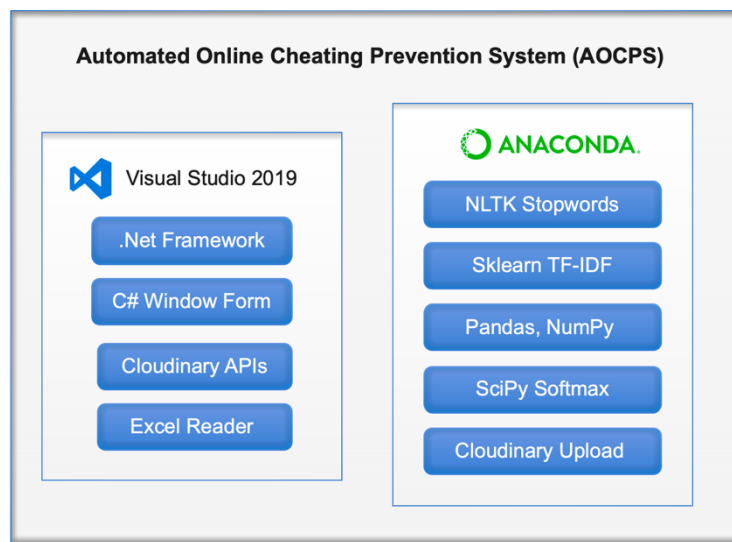


Figure 16- Architecture Diagram for Tools and Techniques

4.2. Client-Side Implementation

4.2.1. C# Window Forms & .Net Framework

.Net is one of the popular developer platforms build with various tools, programming languages and libraries that support to build many different types of applications, running websites, services, desktop applications on windows.

The architecture of the .Net framework is have two major components i.e., CLR and class libraries. All those applications written in different supportive .net languages such as c#, F# and visual basic is compiled into common intermediate language (CIL) then compiled code is stored in assemblies .dll or .exe files. When application runs the common language runtime (CLR) takes above assembly code and uses just-in-time-compiler (JIT) to run into machine code which executes in specific architecture of the computer.

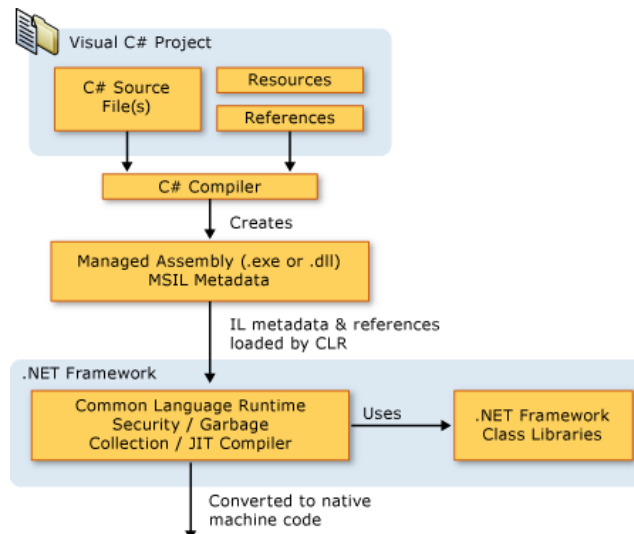


Figure 17-.Net Framework and DesignSome major features

Some major features of .Net / C# Window form are:

Rich interactive UI: The C# window form feature of the .Net framework provides the set of development feature including the graphics, control, data binding, drag-drop features. Also, the set of merged libraries such as reading and writing to the system files. The window forms are created with various controls of displaying the data and accept the user input that communicate with remote computers over network. This technology is useful in our proposed framework to show examination paper in the UI and store meaningful data or information in the form of files which utilizes in generating results easily.

Display and Manipulate Data: Window forms provide the flexible control to show the data in any format. The

form control functions such as on-click actions to retrieve and sent data with various expose methods. The data manipulation in our proposed framework is useful with external cloudinary APIs to send the data files on cloud server. The window forms libraries such as excel reader and writer utilizes to store the data files locally in user's system. The application setting feature addresses the requirement of saving the data file to default locations.

Deploy Apps to Client Computer: Once client desktop application is built then it must send to the users to install and run on their system. The build-in libraries to generate the .exe files are helpful to fulfill the requirement. This feature also helps in our proposed framework as the client users are students and this application is intended to install in multiple operating systems.

Tool Support: As the .Net framework is most popular due to above few features and have largest user community and tool support.

Flexibility: The tool is highly customizable as per the scalability of the application.

4.2.2. Generated Data Files

The front-end application once installed and executed the background process and keystrokes are monitored and two files generated in the user directory. Once these are saved in the local user's directory then uploaded on the cloud server explained in section 2.

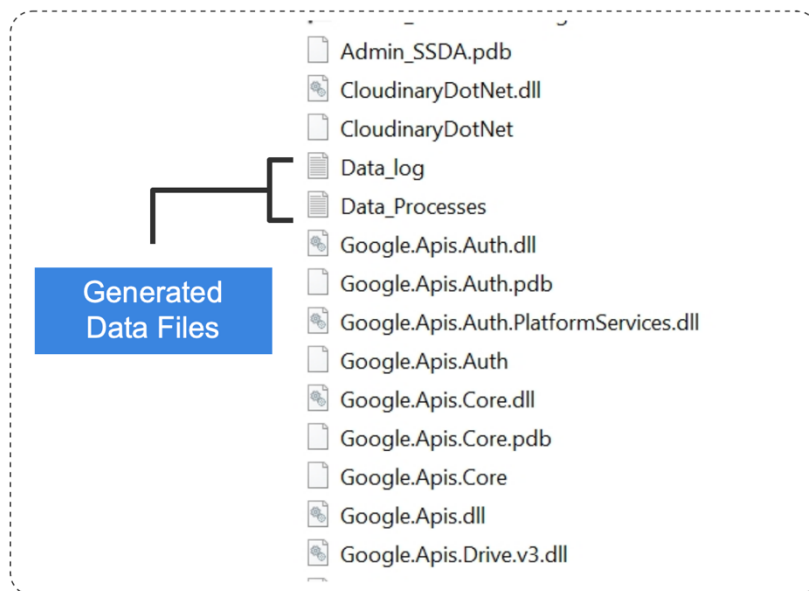


Figure 18- Dataset Generated files

4.2.3. Authentication Feature

In this feature, once user has installed the application successfully the login screen will be shown to authenticate the student. There are many authentication methods available which are used in literature but to achieve the objective of framework i.e., to conduct the cost effective and quick authentication student ids and passwords are pre generated by administration and provided to student via emails.

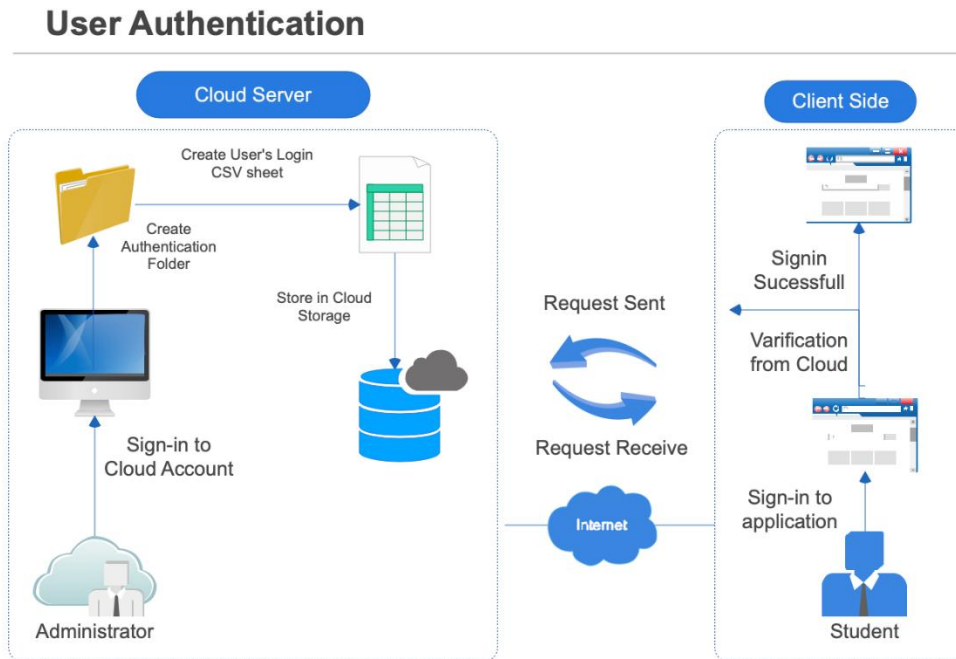


Figure 19- User Authentication Flow

4.2.4. Visualization of Realtime Question Bank

In this step, when user has successfully logged in to the application, an interface will be shown to attempt the exam. Here all the MCQs based questions was created by administrator and available on cloud storage. Here some validations are applied in designing the framework such as in case examination time exceeds the allotted time then a mechanism should be available to submit the examination in given time frame. For this purpose, timer function is used which is available from C# library, which will be enabled to monitor the execution time of the exam.

Question Bank Generation Architecture

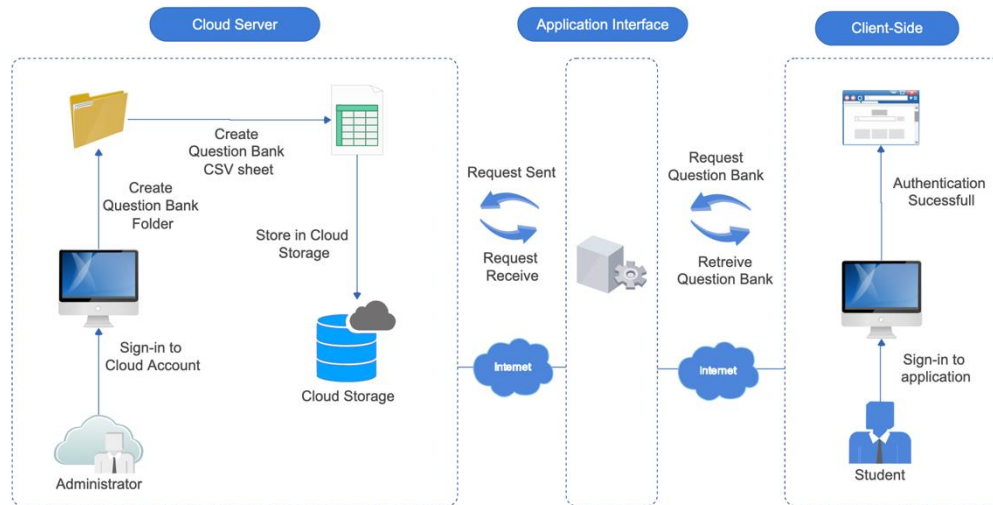


Figure 20- Question Bank Generation Flow

4.2.5. Keylogging

The objective of the proposed automated cheating prevention framework is based on this feature. The automated framework is designed in such a way to detect the user keystrokes hiddenly as soon

Key logs Generation Flow

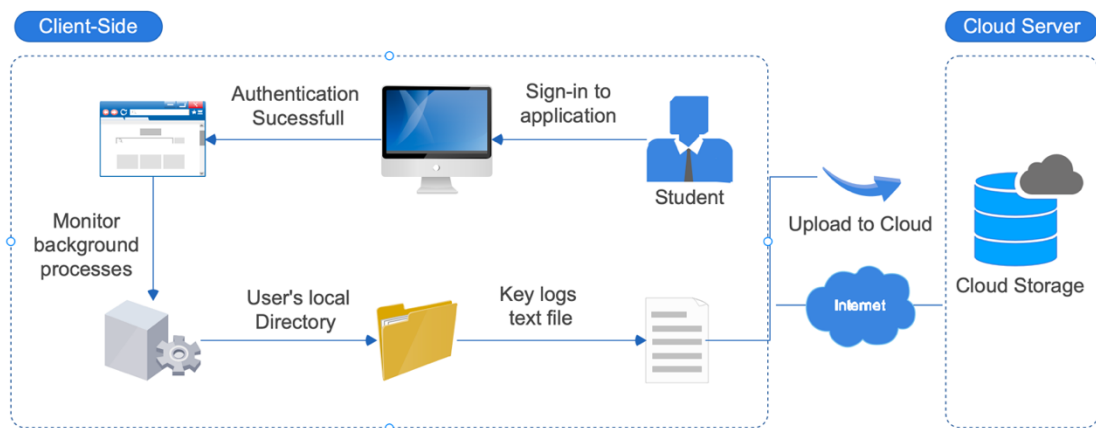


Figure 21-Key Logs Generation Flow

as user start the examination, tracking gets started. The approach is designed with multiple perspective such as student should not be aware of tracking or monitoring activity on the operating system while attempting the question paper. Entire user's keyboard is captured while attempting the question paper.

The approach is designed for below concerns:

- What if user type invalid or special characters which are not useful in predicting the cheating of the student?

- Does key logs captured of only current date or when examination is being conducted?
- How does the key logs only captured of certain time frame while student is attempting the paper?
- How record of real time key logs generated and saved for future evaluation?

To ensure all the above concerns, the framework is designed to mitigate the chances of inaccurate and incomplete data should not be collected. Also, the approach is implemented to utilize the maximum prediction algorithm performance.

4.2.6. System Application History Monitoring

As discussed in above section user's keystrokes are logged hiddenly and stored in a local user's directory, likewise user's applications which are being used during the examination attempt or tracked hiddenly. The monitoring of application is also play important role in predicting the cheating probability during examination. There is many software available that prevent the user to open the browsers, restriction on closing or minimizing the window, but the proposed framework is designed to let user use all the application such as, browsers, communication or chatting desktop application, Skype, Microsoft office, WhatsApp, massager apps and many others.

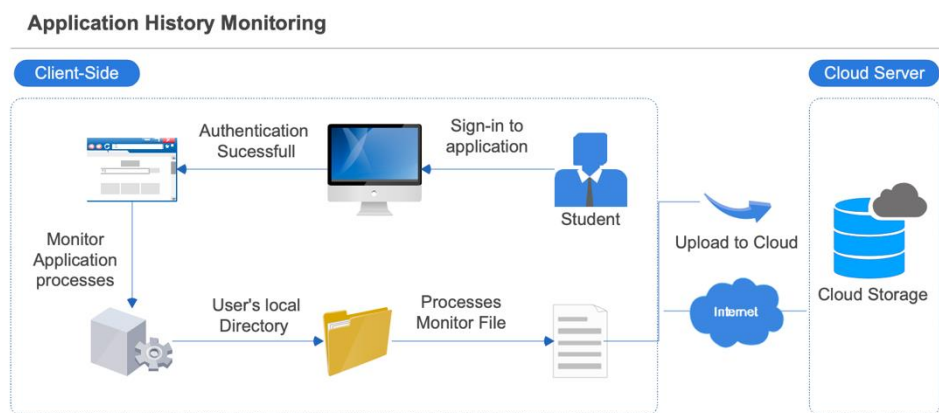


Figure 22- Application History Monitoring

Another concern raised about the background processes running on user's system

The following concerns are facilitated while designing the approach for application history monitoring.

- The tracking of the application will only be tracked while user is attempting the question paper. As soon as user completed the examination and submit the paper successfully the application tracking will be logged off.

- In case, user has already opened many applications which are mentioned in below section then function is designed that trim out old application processes which are being used in background processes on user's system. only count current date and time applications on which examination is being conducted.
- By implementing the above rule, the dataset is more precise and increases the performance of the prediction algorithm.
- In case, user open and close the application multiple time then code is designed to track it properly and maintain the logs, for this purpose a logging loop cycle run after every one minute to refresh the application history and add or replace the concurrent event and logged into application history.
- The monitoring time of all the processes is also logged that help in predictive analysis and achieve the aim of the proposed solution.
- To cutdown the number of applications and browsers, the approach is being used by selecting specific number of browsers and applications which are listed below.

Also, the approach is designed in such a way that above rule which is selection of specific number of applications and browsers could be scaled up in future.

- Specific browser includes, chrome, Safari, Firefox, Internet explorer, Opera and Microsoft edge.
- Communication or chatting application includes, Skype, WhatsApp, Microsoft PowerPoint, PDFs, MS word, MS excel.

4.2.7. Network Infrastructure Handling

4.3. Server-Side Implementation

We have handled the network dysconnectivity of internet during online examination on the client side by using the Microsoft Network API, as soon as network is disconnected the student shall not be able to submit the exam while monitoring is ongoing in background as mentioned in above sections.

4.3.1. Anaconda Python

The anaconda python is the open-source tool popular worldwide for building and training the machine learning algorithms. Our proposed methodology required some free open-source libraries to build the model which can process and generated the required results. For this purpose, various open-source packages are available such as to clean the data and pre-processing purpose pandas and NumPy libraires are used to manipulate the data into data frames and operate numeric calculations discussed in section 3.

Now, anaconda python tool will utilize data set files which are generated from front-end application and generate the prediction by using various python libraries such as pandas, NumPy, SoftMax, file readers

etc. The results generate from key similarities are used to generate the final prediction and results are uploaded on the cloud storage with student name and roll number in the form of CSV sheets. Below prediction algorithm code example is shown which is described in section 3.

4.3.2. Fetch Dataset files & Utilize

The dataset generated files which are explained in above section 4.2.2. The administrator will download the files from cloud server and store it in local directory of python folders where it will be utilized to run the prediction algorithm.

4.3.3. Algorithm

```

396
397 BroSim = sum(Avg_Search_similarity) / len(Avg_Search_similarity) * 0.6
398 KeySim = sum(Avg_key_similarity) / len(Avg_key_similarity) * 0.4
399 dic = \
400     [{"Search \ Key Stoke Data": ['Total cheating Prediction (Key Logs & Browser History)
401     , 'Cheating according to Browsing history',
402     'Cheating according to Keylogs prediction'],
403     'Percentage %': ['80% out of 100%', '60% out of 80%',
404     '40% out of 80%'], 'cheating_Prediction': [(BroSim + KeySim)
405     * 0.8, BroSim, KeySim]}]
406 sim_df = pd.DataFrame(dic)
407 sim_df
408
409 Total_App = (Skypecount + notepadcount + WINWORDcount + whatsappFlag
410             * 2) * 4
411 App_cheating = \
412     pd.DataFrame({'Cheating of other Apps ': 'Notepad/PowerPoint/Skype/whatsapp'
413                 , 'Percentage %': '20 out of 100',
414                 'cheating_Prediction': [Total_App]})
415 App_cheating
416
417 FinalPrediction = (BroSim + KeySim) * 0.8 + Total_App
418
419 cheating_prediction = \
420     pd.DataFrame({'Cheating Source': [' keys of Keyboard / Internet Browser'
421                                     , 'Notepad/PowerPoint/Skype/whatsapp',
422                                     'Total Cheating Prediction'],
423                 'Percentage %': ['80 out of 100', '20 out of 100',
424                 '100 out of 100'], 'cheating_Prediction': [(BroSim
425                 + KeySim) * 0.8, Total_App, FinalPrediction]})
426 cheating_prediction

```

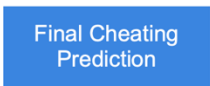


Figure 23- Fetch Dataset File & Utilization

The final cheating prediction is calculated form key similarities of different vectors explained in section 3.1. Here example code is shown in below Figure.

Once the final prediction is calculated then the results are uploaded to cloud storage with student name and roll number in the form of CSV sheets.

4.4. Tool Interface

As discussed in section 2.2.1, application will be installed and user open the application, login interface shown in figure. Users enter name and roll number (password) which is provided by administrator through email and click login.

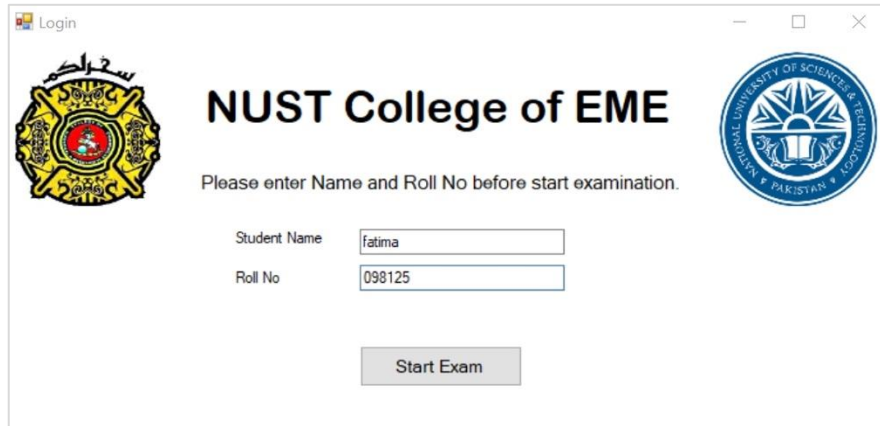


Figure 24- Client-Side Application Interface

Once user is login successfully then question (Mcqs) window is shown to user as explained in above sections for question bank visualization and interface is shown below. Here allotted time count for examination started as soon as user is shown with questions and click on “Submit Test” the data files are saved in local directory and uploaded to cloud server explained in section 2.2.4.

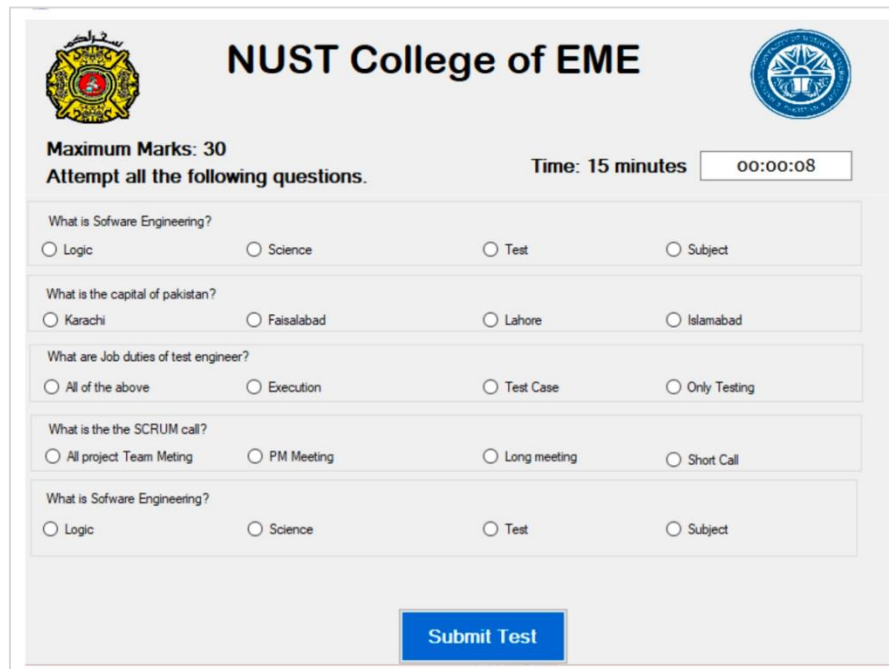


Figure 25- Application Interface II

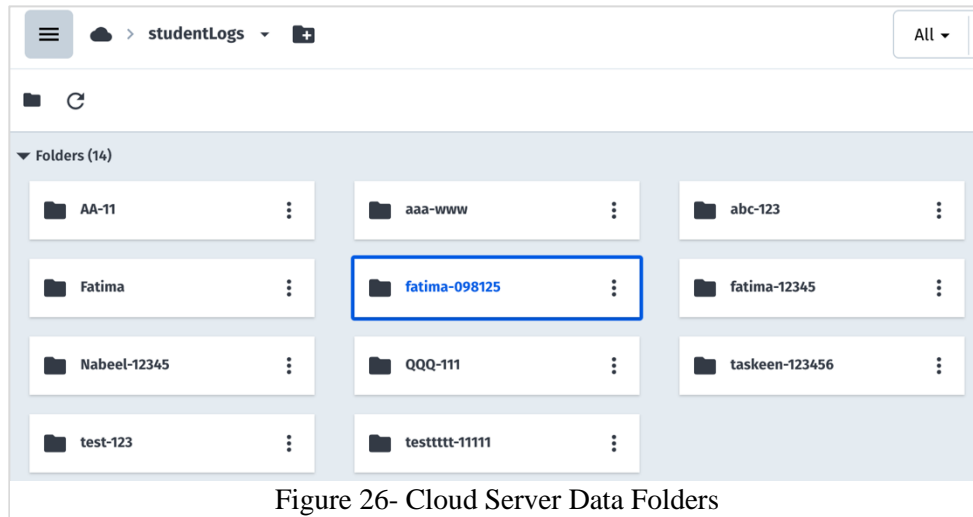


Figure 26- Cloud Server Data Folders

The uploaded files will be saved in cloud storage with student name and roll number as shown above in figure. From here administrator will download the data set files and will use in processing the cheating prediction as explained in section 2.3.1. Now the python script will be converted into executable file to make the administrator side interface user friendly. For this purpose, Pyinstaller tool is used that converts the python scripts into executable file.

api-ms-win-crt-runtime-l1-1-0.dll	11/2/2021 12:09 AM	Application extension
api-ms-win-crt-stdio-l1-1-0.dll	11/2/2021 12:09 AM	Application extension
api-ms-win-crt-string-l1-1-0.dll	11/2/2021 12:09 AM	Application extension
api-ms-win-crt-time-l1-1-0.dll	11/2/2021 12:09 AM	Application extension
api-ms-win-crt-utility-l1-1-0.dll	11/2/2021 12:09 AM	Application extension
base_library	11/2/2021 12:07 AM	WinRAR ZIP archive
keystokesimilarity	11/2/2021 12:09 AM	Application
keystokesimilarity.exe.manifest	11/2/2021 12:09 AM	MANIFEST File
lib_arpack-.OUTLRC34XR23CPDVNJC4LNEGP6P...	11/2/2021 12:09 AM	Application extension
lib_blas-su.YRKGGVITYTJAYNBJNOGROZCFCCMK...	11/2/2021 12:09 AM	Application extension
lib_dop-f2p.3HWIZKD75YUXVW7GYVDBE4FCD...	11/2/2021 12:09 AM	Application extension
lib_test_fo.JF5HTWMUPBXWGWAYEBVEJU3OZAH...	11/2/2021 12:09 AM	Application extension
libansari.R6EA3HQP5KZ6TAXU4Y4ZVTRPT7UVA...	11/2/2021 12:09 AM	Application extension
libbanded5x.2OH54T7PXCLH7KMZFQXI6WES7...	11/2/2021 12:09 AM	Application extension

Figure 27-Python Script Folder

Now data files will be collected by administrator and pasted in the python director folder “content” and administrator will run the application “keystrokesimilarity”.

```

C:\Windows\System32\cmd.exe
0. 0. 0. 0. 0. 0.]

+++++ All similarities of Keys with 30 Questions +++++

Average key_similarity against each Question:
[0.02483215 0.03770605 0.38318085 0.          0.          0.
0.          0.          0.          0.          0.          0.
0.          0.          0.          0.02483215 0.03770605 0.38318085
0.          0.          0.          0.          0.          0.
0.          0.          0.          0.          0.          0.          ]

+++++ Combine similarity of 30 questions are +++++
[0.02483215 0.03770605 0.38318085 0.          0.          0.
0.          0.          0.          0.          0.          0.
0.          0.          0.          0.02483215 0.03770605 0.38318085
0.          0.          0.          0.          0.          0.
0.          0.          0.          0.          0.          0.          ]

*** Total cheating prediction against each question according to search history

File uploaded to server :: {'asset_id': '9f7b99635a56f0932b22e7017c881514', 'version': '1642946158', 'version_id': 'fc3208658577eade1bcd8c6e2f50db8e300b437c718143e8', 'resource_type': 'raw', 'created_at': '2021-10-23T15:29:40', 'etag': 'df209b22b885036317e5833a6d44bda1', 'placeholder': False, 'url': 'raw/upload/v1642946158/processs/students/%5B%27fatima%2012345%27%5D.csv', 'security': 'raw/upload/v1642946158/processs/students/%5B%27fatima%2012345%27%5D.csv', 'api_key': ['fatima 12345'], 'api_key': '124261996441551'}

```

Figure 28- Executed Python Script

administrator run the python application results are generated and uploaded to cloud storage with student name and roll number. As discussed in section 2.3 and 3, the proposed framework techniques will be executed by running this python executable file. The python code is made executable by using tool Pyinstaller. The cheating prediction user interface is made more user friendly and results generation logs are also be shown during the execution of the python code.

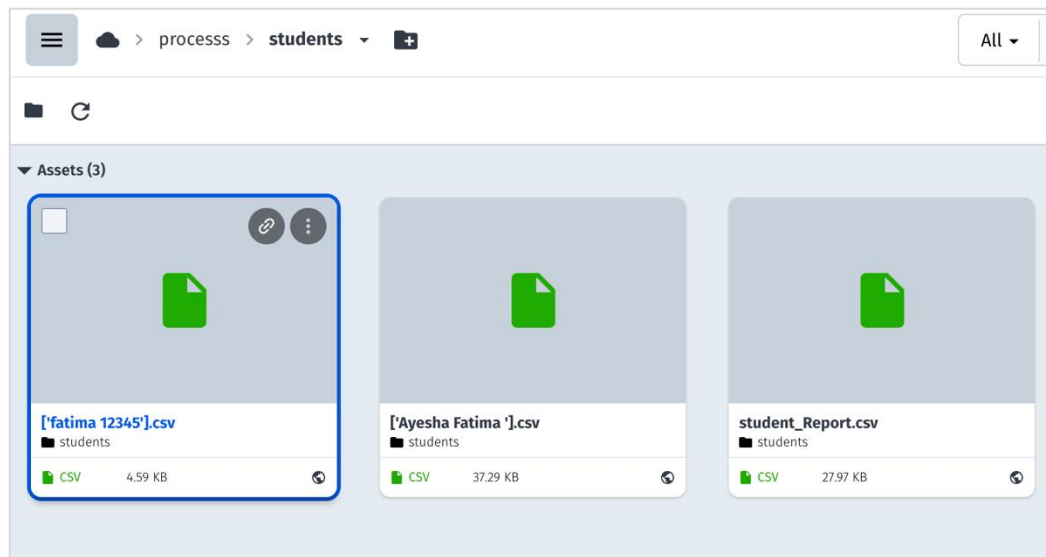


Figure 29- Automated Cheating Prediction Report on Cloud Server

Upon executing the python application acknowledgment is shown “file uploaded to server” and can be found by administrator on cloud account as shown below. Also, administrator can download the result file which is uploaded automatically from python executable file.

The anaconda python is the open-source tool popular worldwide for building and training the machine learning algorithms. Our proposed methodology required some free open-source libraries to build the model which can process and generated the required results. For this purpose, various open-source packages are available such as to clean the data and pre-processing purpose pandas and NumPy libraires are used to manipulate the data into data frames and operate numeric calculations discussed in section 3.

Now, anaconda python tool will utilize data set files which are generated from front-end application and generate the prediction by using various python libraries such as pandas, NumPy, SoftMax, file readers etc. The results generate from key similarities are used to generate the final prediction and results are uploaded on the cloud storage with student name and roll number in the form of CSV sheets. Below prediction algorithm code example is shown which is described in section 4.3

Chapter 5

Validation

CHAPTER 5: VALIDATION

This section presents the validation of our proposed framework with the help of collecting sample data at Saudi Electronic University (SEU). Section 5.1 discusses the validation procedure and results. Section 5.2 and its subsection discusses the in-detail validation.

5.1. Validation Process

Our framework is validated from the Saudi Electronic University (SEU). We have selected 12 different windows operating system and installed the software application through executable file in each system. Now we have taken 12 students to attempt the exams on each system for two subjects mentioned in below table 5.1. From each student we have collected 2 files. One is keystrokes data file and other file is containing user's system processes and browsing history.

Table 13- Exam Project Sample Details

Sr #	Name of Subjects	No of Students	Cheating Class	Files (Per Student)
1	Database	11	N = Non-cheating Case	2
2	Networking	11	Y = Cheating Case	2

Now further we have evaluated our framework by analyzing each student to attempt cheating in one subject and in other to not attempt the cheating. In this way we have compiled the results against each subject which are explained in section 5.2.

5.2. Test Project Details

5.2.1. Non-cheating test scenario with Database subject:

The first five student out of eleven are asked to open the exam "Database" and the total time is 15 minutes. In this subject the student is asked to not attempt any cheating i.e., opening any application or browse on internet and submit the exam. In this case we have received both files "Key logs" which is containing the keystrokes and "process" file which contain all the processes of the system. Here below applications are already closed during exam attempt.

- Skype
- WhatsApp
- MS PowerPoint

- Notepad / PDF

The final computation of the of the cheating score is calculated by the script. We have TF-IDF score calculation in the code that predicts and give final score then further results are converted into percentage as shown below.

```
import nltk
from nltk.corpus import stopwords
nltk.download('stopwords')
from sklearn.feature_extraction.text import TfidfVectorizer

[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data] Unzipping corpora/stopwords.zip.

stop_words_l=stopwords.words('english')

# Combine both Data "Question" and "Keyboar Data"
KeyDocument = pd.concat([Key_data, Question_data] , axis =0)
# removing special characters and stop words from the Keyboard Data
KeyDocument['documents_cleaned']=KeyDocument.Data.apply(lambda x: " ".join(re.sub(r'^a-zA-Z', ' ',w)

# TF-IDF for "Question" and "Keyboar Data"
tfidfvectoriserKey=TfidfVectorizer()
tfidfvectoriserKey.fit(KeyDocument.documents_cleaned)
tfidf_vectors_key=tfidfvectoriserKey.transform(KeyDocument.documents_cleaned)

Key_similarities=np.dot(tfidf_vectors_key,tfidf_vectors_key.T).toarray()
```

In this function “Tfidfvectorizer ()”, previously cleaned data is used and first step to calculate the prediction is by removing the stop words from both datasets (Keystrokes & processes). Then both datasets are combined and then removing the special characters from arrays. Then TF-IDF is calculated and at the last final value is converted into probabilities through function called “SoftMax” as shown below:

```
SearchResult =[]
KeyResult=[]
# convert list of search similarity into a list of probabilities
Search_result = softmax(Avg_Search_similarity)
Key_result = softmax(Avg_key_similarity)
print("**** Total cheating prediction against each question according to search history and keyboard keys *** \n")
for i in range(30):
    index = i+1
    SearchResult.append([index,Question_data.iloc[i],[0] , Search_result[i]])
for i in range(30):
    KeyResult.append(Key_result[i])

Qcheating_Prediction = pd.DataFrame(SearchResult, columns=["Question_No" ,"Question_Details" , "Search_cheating_%"])
Qcheating_Prediction["Keys_cheating_%"] = KeyResult
Qcheating_Prediction.head()
```

Prediction Results:

Our framework has found none of the process from above list. Also, irrelevant keystrokes are detected as shown below:

```
Process : Admin_SSDA , Title: Exam Form , Process Time : 0 Days, 0 Hours, 5 Minutes, 57 Seconds and 799 Milli
Process : SystemSettings , Title: Settings , Process Time : 0 Days, 13 Hours, 9 Minutes, 35 Seconds and 989 M
Process : ApplicationFrameHost , Title: Settings , Process Time : 0 Days, 13 Hours, 9 Minutes, 35 Seconds and
Process : TextInputHost , Title: Microsoft Text Input Application , Process Time : 0 Days, 0 Hours, 17 Minute
Abdullatif s190130995
Process : Admin_SSDA , Title: Exam Form , Process Time : 0 Days, 0 Hours, 6 Minutes, 34 Seconds and 760 Milli
Process : SystemSettings , Title: Settings , Process Time : 0 Days, 13 Hours, 10 Minutes, 12 Seconds and 909 M
Process : ApplicationFrameHost , Title: Settings , Process Time : 0 Days, 13 Hours, 10 Minutes, 12 Seconds an
Process : TextInputHost , Title: Microsoft Text Input Application , Process Time : 0 Days, 0 Hours, 18 Minute
Abdullatif s190130995
Process : Admin_SSDA , Title: Exam Form , Process Time : 0 Days, 0 Hours, 7 Minutes, 3 Seconds and 559 Milli
Process : SystemSettings , Title: Settings , Process Time : 0 Days, 13 Hours, 10 Minutes, 41 Seconds and 721 M
Process : ApplicationFrameHost , Title: Settings , Process Time : 0 Days, 13 Hours, 10 Minutes, 41 Seconds an
Process : TextInputHost , Title: Microsoft Text Input Application , Process Time : 0 Days, 0 Hours, 18 Minute
Abdullatif s190130995
Process : Admin_SSDA , Title: Exam Form , Process Time : 0 Days, 0 Hours, 7 Minutes, 29 Seconds and 454 Milli
Process : SystemSettings , Title: Settings , Process Time : 0 Days, 13 Hours, 11 Minutes, 7 Seconds and 603 M
Process : ApplicationFrameHost , Title: Settings , Process Time : 0 Days, 13 Hours, 11 Minutes, 7 Seconds and
Process : TextInputHost , Title: Microsoft Text Input Application , Process Time : 0 Days, 0 Hours, 19 Minute
```

```
[ 5/9/2022 7:34:49 AM ]DFSDFSDF
```

Here are the results calculated based on dataset files. In all of the students our framework detected minor value for cheating that may cause through any process. Below table is shown with cheating score.

Table 14- Non-Cheating Case Results

No of Students	Subject	Cheating Class N = Non-Cheating Case	Cheating Score
1	Database	N	0
2	Database	N	0.005175358
3	Database	N	0
4	Database	N	0
5	Database	N	0

5.2.2 Cheating Test Scenario with Database subject:

The other six student out of eleven are asked to open the exam “Database” and the total time is 15 minutes. In this subject the student is asked to attempt cheating i.e., opening any application (notepad, skype, PowerPoint, WhatsApp) or browse on internet and submit the exam. In this case

we have received both files “Key logs” which is containing the keystrokes and “process” file which contain all the processes of the system. Here below applications are opened during exam attempt.

- Skype
- WhatsApp
- MS PowerPoint
- Notepad / PDF

The final computation of the of the cheating score is calculated by the script. We have TF-IDF score calculation in the code that predicts and give final score then further results are converted into percentage as shown below.

```
import nltk
from nltk.corpus import stopwords
nltk.download('stopwords')
from sklearn.feature_extraction.text import TfidfVectorizer

[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data] Unzipping corpora/stopwords.zip.

stop_words_l=stopwords.words('english')

# Combine both Data "Question" and "Keyboar Data"
KeyDocument = pd.concat([Key_data, Question_data] , axis =0)
# removing special characters and stop words from the Keyboard Data
KeyDocument['documents_cleaned']=KeyDocument.Data.apply(lambda x: " ".join(re.sub(r'^a-zA-Z',' ',w)

# TF-IDF for "Question" and "Keyboar Data"
tfidfvectoriserKey=TfidfVectorizer()
tfidfvectoriserKey.fit(KeyDocument.documents_cleaned)
tfidf_vectors_key=tfidfvectoriserKey.transform(KeyDocument.documents_cleaned)

Key_similarities=np.dot(tfidf_vectors_key,tfidf_vectors_key.T).toarray()
```

In this function “Tfidfvectorizer ()”, previously cleaned data is used and first step to calculate the prediction is by removing the stop words from both datasets (Keystrokes & processes). Then both datasets are combined and then removing the special characters from arrays. Then TF-IDF is calculated and at the last final value is converted into probabilities through function called “SoftMax” as shown below:

```

SearchResult =[]
KeyResult=[]
# convert list of search similarity into a list of probabilities
Search_result = softmax(Avg_Search_similarity)
Key_result = softmax(Avg_key_similarity)
print("**** Total cheating prediction against each question according to search history and keyboard keys **** \n")
for i in range(30):
    index = i+1
    SearchResult.append([index,Question_data.iloc[i,][0] , Search_result[i]])
for i in range(30):
    KeyResult.append(Key_result[i])

Qcheating_Prediction = pd.DataFrame(SearchResult, columns=["Question_No" ,"Question Details" , "Search_cheating_%"])
Qcheating_Prediction["Keys_cheating_%"] = KeyResult
Qcheating_Prediction.head()

```

Prediction Results:

Our framework has found all the process from above list. Also, relevant keystrokes or browsing history is found in key log data which Matches the question paper as shown below:

```

Process : ApplicationFrameHost , Title: Settings , Process Time : 1 Days, 8 Hours, 17 Minutes, 59 Seconds an
Process : Teams , Title: Calendar | Microsoft Teams , Process Time : 1 Days, 8 Hours, 4 Minutes, 39 Seconds
Process : Admin_SSDA , Title: Exam Form , Process Time : 0 Days, 0 Hours, 0 Minutes, 13 Seconds and 339 Mill
Process : TextInputHost , Title: Windows Input Experience , Process Time : 1 Days, 8 Hours, 17 Minutes, 49 S
Process : WINWORD , Title: Roles and Responsibilities Table - Protected View - Word , Process Time : 0 Day
Process : Notepad , Title: Untitled - Notepad , Process Time : 0 Days, 0 Hours, 20 Minutes, 27 Seconds and 7
Process : Notepad , Title: DOC - Notepad , Process Time : 0 Days, 0 Hours, 2 Minutes, 24 Seconds and 21 Mill
Process : POWERPNT , Title: test powerpoint - PowerPoint , Process Time : 0 Days, 0 Hours, 17 Minutes, 49 Se
Process : chrome , Title: WhatsApp - Google Chrome , Process Time : 1 Days, 6 Hours, 42 Minutes, 47 Seconds
Process : SystemSettings , Title: Settings , Process Time : 0 Days, 1 Hours, 22 Minutes, 10 Seconds and 578
Process : msedge , Title: Document 5.pdf - Work - Microsoft Edge , Process Time : 1 Days, 8 Hours, 17 Minute
Process : EXCEL , Title: results (version 1) - AutoRecovered - Excel , Process Time : 0 Days, 2 Hours, 55
Process : lync , Title: Skype for Business Basic , Process Time : 0 Days, 0 Hours, 21 Minutes, 28 Seconds an

```

```

[ 05/07/2022 2:05:51 am ]WHAT Space DOES Space Capital RDBMS Space
Capital Return Capital IB Back MBD2 Return C Capital OLLECTION Space F
Space Back Back OF Space TABLES Return WHICH Space I Back O Space Back
F Space HE Back Back THE Space FOLLOWING Space IS Space NOT Space THE
Space EXAMPLLE Space OF Space Capital DBMS Return LControlKey| A Capital
H Back WHAT Space IS Space DATA Space CALLED Return

```

As you can see the processes are monitored in above list are fetched from system through our script and saved in a file. The pre-processed data is shown below.

	Data						
0	WhatsApp						
1	Document 5.pdf						
2	what does rdbms consists of						
3	IMB2						
4	which of the following is not the example of DBMS						

Also, all keystrokes are stored and after data pre-processing it is found that keystrokes and browsing history have similarity with the exam questions. Hence based on that the overall average cheating score is shown in below table.

Table 15- Cheating Case Results

No of Students	Subject	Cheating Class Y = Cheating Case	Cheating Score
1	Database	Y	17.6543
2	Database	Y	11.349
3	Database	Y	12
4	Database	Y	12
5	Database	Y	12
6	Database	Y	0

Now, our framework has tested the offline monitoring module which is handled on client side. To validate this function, we have asked three students to disconnect the internet during examination and try to submit. As soon as students perform this action our framework has handled the internet connectivity method as shown in below figure.

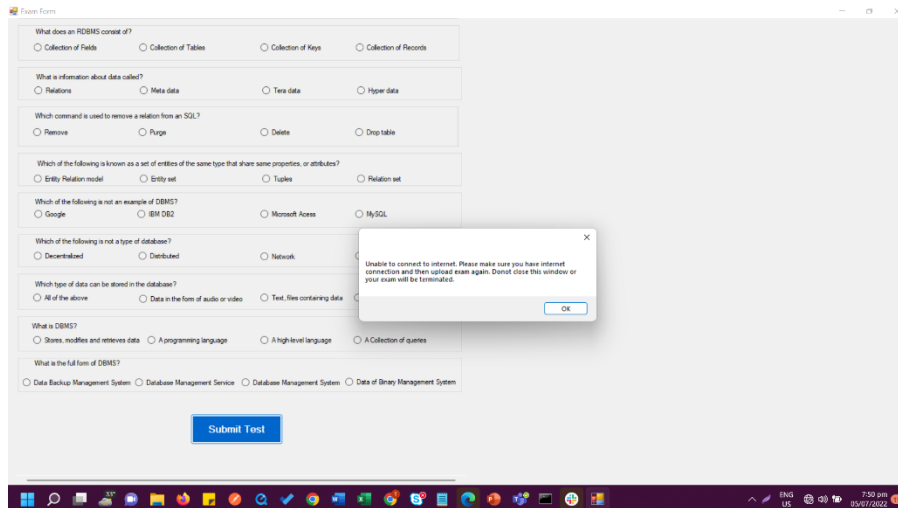


Figure 30- Network Connectivity Handling

The below function checks the connectivity of at the user’s system and if internet connection is not connected then show a pop-up message to connect the internet and then try to submit as shown in figure 30.

```

static Ping p = new Ping();
static string host = "8.8.8.8";
static byte[] buffer = new byte[32];
static int timeout = 1000;
static PingOptions po = new PingOptions();
static PingReply pr;
static bool load = false;
public bool checkConnection()
{
    pr = p.Send(host, timeout, buffer, po);
    return pr.Status == IPStatus.Success;
}

public async void connectionTask()
{
    while (!checkConnection())
    {
        await System.Threading.Tasks.Task.Delay(200);
    }
}

```

Figure 31- Internet Connectivity Function

All the above process mentioned in 5.2.1 and 5.2.2 is repeated with “Network” subject for both cheating and non-cheating cased. Also, results are shown in below table.

Table 16- Prediction Results for Cheating and Non-Creating Cases

No of Students	Subject	Cheating Class N = Non-Cheating Case Y = Cheating Case	Cheating Score
1	Network	N	0
2	Network	N	0.36
3	Network	N	0
4	Network	N	0
5	Network	N	0
6	Network	Y	12.36757895
7	Network	Y	12.36
8	Network	Y	0
9	Network	Y	12.34971429
10	Network	Y	12.34971429
11	Network	Y	0

To validate our claim that accurate and high accuracy cheating prediction results, we have calculated the accuracy, precision and recall of final cheating results detected by our framework. To calculate all the measurements, we have used their generic formulas as explained below.

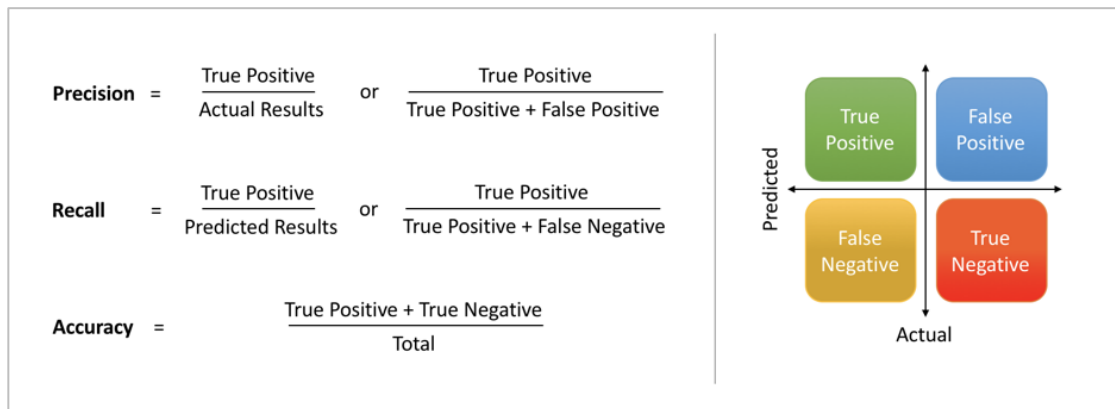


Figure 32- Precision, Recall & Accuracy Formula

Accuracy: The Accuracy measure we have used to validate that at what extent framework predict the correct results. As from above formula we know that the sum of true positive and true negative will be calculated and divided by the total numbers of samples. Hence this formula will give us the numbers how accurate our framework predicts cheating.

$$\text{Accuracy} = \frac{\text{True Positive} + \text{True Negative}}{\text{Total Samples}}$$

Precision: The precision metrics tell us how accurate the model or framework out of all the predictive positive how many out of these are actual positive. Thus, this metrics is good measure to determine when we have the false positive high and low. Below formula calculate the precision.

$$\text{Precision} = \frac{\text{True Positive}}{\text{Total Predicted Results}}$$

Recall: This metrics tell us how many of the actual positive in our model is detecting when we have labeled it as positive (True Positive). Below formula calculate the recall.

$$\text{Precision} = \frac{\text{True Positive}}{\text{Total Actual Results}}$$

The results calculated from above formulas are shown in table. The analysis is shown that our framework has predicted 90% accuracy, 100% Precision and 83% recall. Below table is shown with all testing projects results.

Table 17- Total Prediction Results in Percentage

Project	Metrics	Score
P1	Accuracy	0.909090
P1	Precision	1.00
P1	Recall	0.833333
P2	Accuracy	0.818181
P2	Precision	1.00
P2	Recall	0.833333

Chapter 6

Discussion and Limitation

CHAPTER 6: Discussion and Limitation

6.1. Discussion

From the research it has been analyzed that in pandemic COVID19, online exam proctoring systems are vastly used by various institutions around the globe and particularly the system adopted by developed countries. The OEM have a lot of benefits such as, time saving, reduce cost for conducting physical exams and use of other resources. In the parallel where all the online examination proctoring solutions are giving benefits where student dis honesty has increased and that created the need to introduce the monitoring solution to overcome this dishonesty and, in this regard, much research have proposed framework which utilizes latest technology algorithms and equipment to reduce the cheating or di honesty of the students. But many factors have made the online examination proctoring solutions very costly, and a lot of hardware equipment are required to adopt the OEM solutions which is particularly difficult to manage for less developed countries. Hence, there is need to propose a cost-effective solution for less developed countries and no tool is available which provide integrate able solution which is also cost effective.

Our open-source framework for anti-cheat online examination in Pakistan has proposed that ensures improved ease of use of software and cheating detecting during offline monitoring and low network bandwidth. The proposed framework used NLP technique to find out the cheating prediction on student's monitoring process and data. The benefit of using the NLP technique to generate the high accurate results as per studies. Now to make the cost-effective solution the framework is developed using cloud technology that also help to monitor the offline processes or activities and ultimately provide the usability of minimum internet bandwidth requirement. We have used several .Net libraries to manage the student system's processes, monitor key logs, question bank visualization, and manage offline monitoring and then generate report for cheating prediction for each student.

Our approach supports cheating prediction of a student during online examination for less developed countries where less bandwidth network is addressed so this adoption factor can be addressed for less developed countries as well as to make cost-effective solution. It also handles exam submission if network is not stable then student will be able to submit exam once network is stable again. The framework is successfully validated using 11 student's real-time dataset.

6.2. Limitation

This proposed framework improves the usability of software application, cost effectiveness and low bandwidth network requirement. The framework provides monitoring of four major applications which can be extended in future to scale the monitoring activities. Also, our framework has used cloud-based authentication of the user and facial recognition can also be implemented to extend the framework in future. Our framework is flexible to scalable and integrate more authentication methods. Currently, our framework is validated with few numbers of student's data. If the framework is validated with large number of students, then results can be validated with higher accuracy and performance of the framework.

Chapter 7

Conclusion and Future Work

CHAPTER 7: CONCLUSION AND FUTURE WORK

an open-source framework for anti-cheat online examination in Pakistan has proposed that ensures improved ease of use of software and cheating detecting during offline monitoring and low network bandwidth. The framework also automated the cheating prediction reports. The proposed framework used NLP technique to find out the cheating prediction on student's monitoring process and data. The benefit of using the NLP technique to generate the high accurate results as per studies. Now to make the cost-effective solution the framework is developed using cloud technology that also help to monitor the offline processes or activities and ultimately provide the usability of minimum internet bandwidth requirement. We have used several .Net libraries to manage the student system's processes, monitor key logs, question bank visualization, and manage offline monitoring and then generate report for cheating prediction for each student. Also, we have used Visual studio tool with .Net framework to develop the front end of the application and python to develop back-end script for prediction.

Our approach supports cheating prediction of a student during online examination for less developed countries where less bandwidth network is addressed so this adoption factor can be addressed for less developed countries as well as to make cost-effective solution. It also handles exam submission if network is not stable then student will be able to submit exam once network is stable again. The framework is successfully validated using 11 student's real-time dataset.

Future work includes implementing other authentication method which are not implemented so far, can be implemented into this framework for more accurate behavior analysis. Also, a greater number of applications can be included to have more cheating prediction reliable results. Currently our framework is limited to most useable four applications.

REFERENCES

1. Kryterion. (2021). Kryterion Global Testing Solutions. [Online]. Available: <https://www.kryteriononline.com/>
2. A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, arXiv:1704.04861. [Online]. Available: <http://arxiv.org/abs/1704.04861>
3. A. A. Jain, A. K. Flynn, and P. J. Ross, Handbook of Biometrics. Springer, 2008. [Online]. Available: <https://www.springer.com/gp/book/97803877110402#aboutBook>
4. ProctorU. (2021). The Leading Proctoring Solution for Online Exams. [Online]. Available: <https://www.proctoru.com/>
5. Examity. (2021). Better Test Integrity. [Online]. Available: <https://examity.com/>
6. PSIONline. (2021). Certification Testing Services and Programs. [Online]. Available: <https://www.psonline.com/en-gb/certification/>
7. ProctorExam. (2021). Infrastructure for Online Proctoring & Invigilation. [Online]. Available: <https://proctorexam.com/>
8. (2021). Assessment Tools for Learning Services. [Online]. Available: <https://web.respondus.com/>
9. RemoteProctor. (2021). Remote Proctor. [Online]. Available: <https://remoteproctor.com/>
10. OnVUE. (2021). OnVUE. [Online]. Available: <https://home.pearsonvue.com/Test-Owner/Deliver/Onlineproctored.aspxBVirtual>. (2021).
11. Online Proctoring Redefined. [Online]. Available: <https://bvirtualinc.com/>
12. L. Verified. (2021). Make Your Online Learning Defensible. [Online]. Available: <https://learnerverified.com/>
13. Proctorio. (2021). A Comprehensive Learning Integrity Platform. [Online]. Available: <https://proctorio.com/>
14. Proctortrack. (2021). Trusted Exam Integrity | Remote Online Proctoring. [Online]. Available: <https://www.proctortrack.com/>
15. Comprobo. (2021). Online Validation. [Online]. Available: <https://comprobo.co.uk/>
16. Sumadi. (2021). AI-Powered Proctoring. [Online]. Available: <https://sumadi.net/>
17. ProctorFree. (2021). Secure Online Proctoring. [Online]. Available: <http://proctorfree.com/>

18. HonorLock. (2021). Honorlock On-Demand Online Proctoring Service. [Online]. Available: <https://honorlock.com/>
19. ExamSoft. (2021). Learning Assessments Tools & Software. [Online]. Available: <https://https://examsoft.com/>
20. Y. Khlifi and H. El-Sabagh, "A novel authentication scheme for e- assessments based on student behavior over e-learning platform," *Int. J. Emerg. Technol. Learn.*, vol. 12, no. 4, pp. 62–89, 2017. [Online]. Avail- able: <https://online-journals.org/index.php/i-jet/article/view/6478>
21. Z. Zhang, M. Zhang, Y. Chang, S. Esche, and C. Chassapis, "A virtual laboratory system with biometric authentication and remote proctoring based on facial recognition," *Comput. Educ. J.*, vol. 7, no. 4, pp. 74–84, 2016.
22. Z. Zhang, E.-S. Aziz, S. Esche, and C. Chassapis, "A virtual proctor with authentication for facilitating distance education," in *Online Engineering Internet of Things*, M. E. Auer and D. G. Zutin, Eds. Cham, Switzerland: Springer, 2018, pp. 110–124.
23. H. S. G. Asep and Y. Bandung, "A design of continuous user verification for online exam proctoring on M-learning," in *Proc. Int. Conf. Electr. Eng. Informat. (ICEEI)*, Jul. 2019, pp. 284–289.
24. L. K. Musambo and J. Phiri, "Student facial authentication model based on OpenCV's object detection method and QR code for Zambian higher institutions of learning," *Int. J. Adv. Comput. Sci. Appl.*, vol. 9, no. 5, Jan. 2018.
25. F. Guillen-Gamez, I. García-Magariño, and G. Palacios, *Comparative Analysis Between Different Facial Authentication Tools for Assessing Their Integration in m-Health Mobile Applications*. Cham, Switzerland: Springer, Mar. 2018, pp. 1153–1161. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-77712-2_110
26. S. Sawhney, K. Kacker, S. Jain, S. N. Singh, and R. Garg, "Real-time smart attendance system using face recognition techniques," in *Proc. 9th Int. Conf. Cloud Comput., Data Sci. Eng. (Confluence)*, Jan. 2019, pp. 522–525.
27. A. Alshbtat, N. Zanoon, and M. Alfraheed, "A novel secure fingerprint-based authentication system for student's examination system," *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 9, pp. 515–519, 2019. [Online]. Available: <https://thesai.org/Publications/ViewPaper?Volume=10&Issue=9&Code=IJACSA&SerialNo=68>
28. J. V. Monaco, J. C. Stewart, S.-H. Cha, and C. C. Tappert, "Behavioral biometric verification of student identity in online course assessment and authentication of authors in literary works," in *Proc. IEEE 6th Int. Conf. Biometrics, Appl. Syst. (BTAS)*, Sep. 2013, pp. 1–8.

29. E. Fior and K. Kowalski, "Continuous biometric user authentication in online examinations," in Proc. Int. Conf. Inf. Technol., Jan. 2010, pp. 488–492.
30. Y. Atoum, L. Chen, A. X. Liu, S. D. H. Hsu, and X. Liu, "Automated online exam proctoring," IEEE Trans. Multimedia, vol. 19, no. 7, pp. 1609–1624, Jul. 2017.
31. A. Okada, I. Noguera, L. Alexieva, A. Rozeva, S. Kocdar, F. Brouns, T. Ladonlahti, D. Whitelock, and A. Guerrero-Roldán, "Pedagogical approaches for e-assessment with authentication and authorship verification in higher education," Brit. J. Educ. Technol., vol. 50, no. 6, pp. 3264–3282, Nov. 2019.
32. G. Fenu, M. Marras, and L. Boratto, "A multi-biometric system for continuous student authentication in e-learning platforms," Pattern Recognit. Lett., vol. 113, pp. 83–92, Oct. 2018.
33. L. Slusky, "Cybersecurity of online proctoring systems," J. Int. Technol. Inf. Manage., vol. 29, no. 3, pp. 56–83, 2020.
34. F. Guillen-Gamez, J. Bravo, and I. García-Magariño, "Students' perception of the importance of facial authentication software in moodle tools," Int. J. Eng. Educ., vol. 33, pp. 84–90, Jan. 2017.
35. A. Ullah, H. Xiao, and T. Barker, "A dynamic profile questions approach to mitigate impersonation in online examinations," J. Grid Comput., vol. 17, no. 2, pp. 209–223, Jun. 2019, doi: 10.1007/s10723-018-9442-6.
36. S. A. Razak, N. H. M. Nazari, and A. Al-Dhaqm, "Data anonymization using pseudonym system to preserve data privacy," IEEE Access, vol. 8, pp. 43256–43264, 2020.
37. L. Li, X. Mu, S. Li, and H. Peng, "A review of face recognition technology," IEEE Access, vol. 8, pp. 139110–139120, 2020.
38. S. Zhang, X. Zhu, Z. Lei, H. Shi, X. Wang, and S. Z. Li, "FaceBoxes: A CPU real-time face detector with high accuracy," in Proc. IEEE Int. Joint Conf. Biometrics (IJCB), Oct. 2017, pp. 297–309. [Online]. Available: <https://www.sciencedirect.com/science/article/abs/pii/S0925231219310719>
39. L. Unzueta, W. Pimenta, J. Goenetxea, L. P. Santos, and F. Dornaika, "Efficient generic face model fitting to images and videos," Image Vis. Comput., vol. 32, no. 5, pp. 321–334, May 2014.
40. X. Liu, X. Ma, J. Wang, and H. Wang, "M3L: Multi-modality mining for metric learning in person re-identification," Pattern Recognit., vol. 76, pp. 650–661, Apr. 2018.
41. F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2015, pp. 815–823. [Online]. Available: <https://ieeexplore.ieee.org/document/7298682>

42. K. Zuiderveld, “Contrast limited adaptive histogram equalization,” in *Graphics Gems IV*. 1994. [Online]. Available: <https://dl.acm.org/doi/10.5555/180895.180940>
43. W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg, *SSD: Single Shot Multibox Detector (Lecture Notes in Computer Science: Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Springer, 2016. [Online]. Available: <https://www.springer.com/gp/book/9783319464770>
44. R. A. Yeh, C. Chen, T. Y. Lim, A. G. Schwing, M. Hasegawa-Johnson, and M. N. Do, “Semantic image inpainting with deep generative models,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6882–6890. [Online]. Available: <https://ieeexplore.ieee.org/document/8100211>
45. I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *Proc. 27th Int. Conf. Neural Inf. Process. Syst. (NIPS)*, vol. 2. Cambridge, MA, USA: MIT Press, 2014, pp. 2672–2680. [Online]. Available: <http://dl.acm.org/citation.cfm?id=2969033.2969125>
46. D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, “The kaldı speech recognition toolkit,” in *Proc. IEEE Workshop Autom. Speech Recognit. Understand. (ASRU)*, Waikoloa, HI, USA. Piscataway, NJ, USA: IEEE Signal Processing Society, Dec. 2011, p. 30. [Online]. Available: <https://dblp.org/db/conf/asru/asru2011.html>
47. L. Wan, Q. Wang, A. Papir, and I. L. Moreno, “Generalized end-to-end loss for speaker verification,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 4879–4883. [Online]. Available: <https://ieeexplore.ieee.org/document/8462665>
48. Q. Wang, C. Downey, L. Wan, P. A. Mansfield, and I. L. Moreno, “Speaker diarization with LSTM,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 5239–5243. [Online]. Available: <https://ieeexplore.ieee.org/document/8462628>
49. Y. Zhong, Y. Deng, and A. K. Jain, “Keystroke dynamics for user authentication,” in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2012, pp. 117–123. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/6239225>
50. H. Crawford, “Keystroke dynamics: Characteristics and opportunities,” in *Proc. 8th Int. Conf. Privacy, Secur. Trust*, Aug. 2010, p. 20108.
51. A. T. Kiyani, A. Lasebae, K. Ali, M. U. Rehman, and B. Haq, “Continuous user authentication featuring keystroke dynamics based on robust recurrent confidence model and ensemble learning approach,” *IEEE Access*, vol. 8, pp. 156177–156189, 2020.
52. A. F. M. N. H. Nahin, J. M. Alam, H. Mahmud, and K. Hasan, “Identifying emotion by keystroke dynamics and text pattern analysis,” *Behav. Inf. Technol.*, vol. 33, no. 9, pp. 987–996, Sep. 2014

53. R. Moskovitch, C. Feher, A. Messerman, N. Kirschnick, T. Mustafić,
54. Camtepe, B. Löhlein, U. Heister, S. Möller, L. Rokach, and Y. Elovici, “Identity theft, computers and behavioral biometrics,” in Proc. IEEE Int. Conf. Intell. Secur. Informat., Jun. 2009, pp. 155–160. [Online]. Available: <https://ieeexplore.ieee.org/document/5137288>
55. L. Xiaofeng, Z. Shengfei, and Y. Shengwei, “Continuous authentication by free-text keystroke based on CNN plus RNN,” Procedia Comput. Sci., vol. 147, pp. 314–318, Jan. 2019.
56. Cloud Computing Elasticity.[Online], <https://github.com/Deivakumaran/Elasticity-in-Cloud-Computing-using-Multithreaded-Programming>, accessed on May 2021.
57. CloudComputing.[Online],https://github.com/absnaik810/CloudComputing/blob/master/Project%201/ahnaik_project1, accessed on May 2021.
58. JLint.[Online], <https://sourceforge.net/projects/jlint/>, accessed on May 2021.
59. Sonarqube.[Online], <https://www.sonarqube.org/>, accessed on May 2021.
60. Facebook Infer.[Online], <https://www.facebookInferorg/>, accessed on May 2021.